Direct data-driven design of LPV controllers with soft performance specifications

Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 359 (2022) 816–836

www.elsevier.com/locate/jfranklin

Direct data-driven design of LPV controllers with soft performance specifications

Max van Meer[a], Valentina Breschi[b,*], Tom Oomen[a], Simone Formentin[b]

[a] Department of Mechanical Engineering, Building 15, Gemini-Zuid, Eindhoven University of Technology, Eindhoven, The Netherlands

[b] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via G. Ponzio 34/5 - 20133 Milano, Italy

Received 29 June 2020; received in revised form 31 March 2021; accepted 22 April 2021

Available online 15 May 2021

Abstract

If only experimental measurements are available, direct data-driven control design becomes an appealing approach, as control performance is directly optimized based on the collected samples. The direct synthesis of a feedback controller from input-output data typically requires the blind choice of a reference model, that dictates the desired closed-loop behavior. In this paper, we propose a data-driven design scheme for linear parameter-varying (LPV) systems to account for *soft* performance specifications. Within this framework, the reference model is treated as an additional *hyper-parameter* to be learned from data, while the user is asked to provide only indicative performance constraints. The effectiveness of the proposed approach is demonstrated on a benchmark simulation case study, showing the improvement achieved by allowing for a flexible reference model.

1. Introduction

Due to the increasing complexity of modern systems and devices and the availability of ever-increasing datasets and computational power, control design techniques relying on

* Corresponding author.

E-mail address: valentina.breschi@polimi.it (V. Breschi).

https://doi.org/10.1016/j.jfranklin.2021.04.052

data have a large potential. By drifting away from more conventional two-stage strategies, consisting of a preliminary identification phase and a subsequent model-based control design task, in the last decades several techniques have been proposed for direct data-driven control design, ranging from well-established approaches, e.g., Virtual Reference Tuning (VRFT) [1], Iterative Feedback Tuning (IFT) [2,3] and Correlation-based Tuning (CbT) [4,5], to the recent advances in data-driven model predictive control [6,7].

At early stages, data-driven techniques focused on the control of linear time invariant (LTI) plants, but recent advances in nonlinear model learning led to their extensions to more complex classes of systems, such as linear parameter varying (LPV) [8,9], hybrid [10] and nonlinear [11] plants. These approaches require the selection of a reference model embedding the desired closed-loop features *before-hand*, which is rather challenging especially for complex systems, if no prior knowledge is available. Indeed, the selection of simple LTI references might impose excessively demanding requirements on the actual closed-loop, which cannot be achieved in practice. At the same time, more complex reference models typically require accurate knowledge of the plant to control, which is usually not the case when direct data-driven approaches are employed.

Instead of looking at the reference model as an input to be fixed a-priori, which is rarely the case in practice, techniques have been introduced to conceive the latter as an additional *degree of freedom* of the design method. These approaches, see e.g., [12–16], require one to solely select the structure of the reference model a-priori, by allowing for the calibration of its parameters within the design phase. Among these methods, the ones presented in [13,15] are specifically designed to handle non-minimum phase plants, while the approach proposed in [16] has proven to be effective even in a nonlinear scenario. As shown in [12], user-defined - typically “soft” - requirements (e.g., minimum acceptable bandwidth, maximum overshoot of the step response, etc.) can be embedded as suitable constraints on the reference model parameters. Due to this fundamental change in perspective, instead of directing the control design towards a fixed reference model, the latter can be shaped so as to achieve the best controller within its class that satisfies the user-defined criteria.

The goal of this work is to propose an approach based on Bayesian Optimization (BO) for the design of LPV controllers for minimum-phase LPV systems. The approach relies on the minimization of a cost accounting for both the performance of the controller and that of the “flexible” reference model similar to the one exploited in [16]. Nonetheless, differently from [16] the objective function does not require the user to prefix the reference signal to be tracked, but it relies on the introduction of a *fictitious* reference signal. By focusing on LTI reference models only, we can incorporate constraints on their parameters that can easily be related to the features of admissible closed-loop responses via standard system theory.

The design of the controller is carried out through the approaches proposed in [8], which handle *single-input single-output* (SISO) systems only, but they allow one to tackle the cases in which the controller structure is selected before-hand or inferred from the available data.

By applying the developed technique to two benchmark problems in a simulation environment, we further assess the performance of the proposed strategy and the effect of different choices for the tunable parameters on the attained closed-loop performance. This last analysis provides the user with additional *practical hints* on the sensitivity of the approach to such choices, which might be useful when considering a different case study.

The paper is organized as follows. The addressed direct data-driven control problem is introduced in Section 2, while the design strategy based on soft specifications and flexible reference models is described in Section 3. In Section 4 the results obtained by employing the proposed design technique on two benchmark simulation examples are reported and throughly commented. Conclusions and directions for future work are finally discussed in Section 5.

2. Setting and goal

Let $\mathcal{G}_p$ be a *single-input single-output* (SISO) minimum-phase LPV system, whose evolution is described by the difference equation

$$\mathcal{G}_p: \quad y_o(t) = \sum_{i=1}^{n_a} a_i(p,t) y_o(t-i) + \sum_{j=0}^{n_b} b_j(p,t) u(t-j), \tag{1a}$$

where $u(t) \in \mathbb{R}$ and $y_o(t) \in \mathbb{R}$ denote the input fed to the system and its noiseless output at time $t \in \mathbb{N}$, respectively. Due to its LPV nature, the behavior of $\mathcal{G}_p$ described in (1a) depends on the measurable signals $p(t) \in \mathbb{P} \subseteq \mathbb{R}^{n_p}$ (the so-called *scheduling variables*). In particular, the *unknown* coefficients $\{a_i(p,t)\}_{i=1}^{n_a}$ and $\{b_j(p,t)\}_{j=0}^{n_b}$ are assumed to be possibly nonlinear dynamic maps of a finite scheduling sequence, which might include the current and past values of the scheduling variables. All signals are assumed to be equal to zero for $t < 0$.

The goal is to design a controller for the plant $\mathcal{G}_p$ so as to achieve some *desired* closed-loop performance, e.g., to satisfy given requirements on the step response. To this end, we assume that open-loop experiments can be performed by supplying the system with persistently exciting inputs $\mathcal{U}_T = \{u(t)\}_{t=1}^{T}$ and scheduling sequence $\mathcal{P}_T = \{p(t)\}_{t=1}^{T}$, and that the corresponding measured outputs $\mathcal{Y}_T = \{y(t)\}_{t=1}^{T}$ are corrupted by an additive, zero-mean, stationary colored noise sequence $\{v(t)\}_{t=1}^{T}$, namely

$$y(t) = y_o(t) + v(t), \quad t = 1, \dots, T. \tag{1b}$$

The resulting dataset $\mathcal{D}_T = \{\mathcal{U}_T, \mathcal{P}_T, \mathcal{Y}_T\}$ is exploited to directly design a controller for $\mathcal{G}_p$, without identifying a model for the plant first. Because of the LPV dynamics of $\mathcal{G}_p$, we focus on the design of LPV controllers, parameterized as

$$\mathcal{K}_p(\theta): u(t) = \sum_{i=1}^{n_{a_k}} a_i^k(p,t,\theta) u(t-i) + \sum_{j=0}^{n_{b_k}} b_j^k(p,t,\theta) e(t-j), \tag{2}$$

where n_{a_k}, n_{b_k} are *fixed* a-priori and $e(t) = r(t) - y(t) \in \mathbb{R}$ denotes the tracking error attained at time t with respect to a certain reference $r(t) \in \mathbb{R}$. For the controller to be as versatile as possible, the coefficients $\{a_i^k(p,t,\theta)\}_{i=1}^{n_{a_k}}$ and $\{b_j^k(p,t,\theta)\}_{j=0}^{n_{b_k}}$ are assumed to be nonlinear dynamic functions of the *unknown* parameter vector θ and the scheduling sequence $\{p(t-h)\}_{h=0}^{m}$, with $m \in \mathbb{N}$ fixed a priori.

According to this choice, the control design problem thus involves learning (i) the parameters θ and (ii) the structure of the controller coefficients, if this is not given. Both these tasks can be handled using the data-driven design methods proposed in [8]. However, these approaches require the user to blindly select a reference model that characterizes the desired closed-loop behavior *before-hand*. This choice is of crucial importance to shape the closed-loop response of the system and, at the same time, it is usually challenging, since it is made with little to no knowledge about the actual plant.

To overcome this limitation, in this paper we propose a change of perspective and treat the reference model as an *hyper-parameter* to be tuned from data. The user is then (more realistically) asked to fix the structure of the reference model and then to provide only "*soft*"

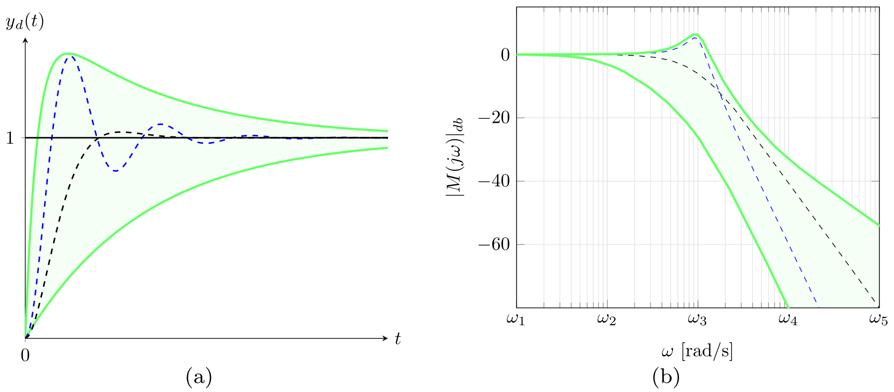


Fig. 1. Acceptable closed-loop ranges provided by the user (green area) against two admissible random reference models (black and blue dashed lines). Examples of soft specifications on (a) the step and (a) the frequency responses. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

specifications, like bandwidth intervals or disturbance rejection levels. The control design procedure then picks up the best admissible closed-loop behavior to find an achievable, yet performing, controller within these bounds (see Fig. 1 for a pictorial view of possible examples associated to the above conception of the reference model).

Even though the approaches presented in [8] allow one to handle both LTI and LPV reference models, in this work we focus on the set of stable LTI reference models

$$\mathcal{M}(\varphi): \quad y_d(t) = M(\varphi, q)r(t), \tag{3}$$

where $y_d(t) \in \mathbb{R}$ is the desired output, q is the shift operator (namely $q^i u(t) = u(t+i)$, $i \in \mathbb{Z}$) and $\varphi \in \mathbb{R}^{n_\varphi}$ is the vector of the reference model parameters. T he use of LTI reference models allows the user to exploit relationships from standard system theory to relate the reference model parameters and the desired closed-loop specifications, thus making it user-friendly. Since $\mathcal{G}_p$ is LPV, we stress that the choice of a suitable LTI reference model is far from trivial, since one might end up requiring excessively demanding inputs and dependences on the scheduling parameter, that are unachievable in practice [8].

Inspired by the strategy proposed in [12], we present a design procedure for LPV controllers with flexible reference models of *customizable* complexity, which involves the optimization of an a-priori selected performance-oriented objective $J : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\varphi} \to \mathbb{R}$ on both the reference model and the controller parameters. In particular, the strategy relies on the solution of the optimization problem

$$\begin{aligned}(\varphi^\star, \theta^\star) = \arg\min\nolimits_{\varphi,\theta} \ & J(\varphi, \theta), \\ \text{s.t.} \ & \varphi \in [\underline{\varphi}, \overline{\varphi}],\end{aligned} \tag{4}$$

where $\underline{\varphi}, \overline{\varphi} \in \mathbb{R}^{n_\varphi}$ are bounds on the reference model dictated by the user, according to the given closed-loop specifications. We stress that, by properly selecting the cost in (4), the design strategy does *not* involve any closed-loop experiment. Moreover, since we rely on a model-reference design technique, the parameters θ of the controller depend on the ones of the flexible reference model, namely $\theta = \theta(\varphi)$.

3. Joint data-driven design of LPV controllers and reference models

In order to learn a LPV controller when the reference model is not fixed but can vary within a certain domain, we embed the approach proposed in [8] within a Bayesian optimization (BO) based routine inspired to the one in [12] for LTI systems. We stress that this choice allows us to avoid naive grid searches over the space of feasible reference models, with this space here explored according to the given performance-oriented cost $J(\varphi, \theta)$.

Consider the *fictitious* reference signal

$$r_v(\varphi, t) = M(\varphi, q)^{-1} y(t), \tag{5}$$

namely the set point that would have generated the measured outputs $\mathcal{Y}_T$ when supplied to $M(\varphi, q)$, with $M(\varphi, q)^{-1}$ denoting the inverse reference model. Moreover, let $u_c(\theta(\varphi), t)$ indicate the input generated by the controller with structure $\mathcal{K}_p(\theta)$ in (2) associated to the reference model $M(\varphi, q)$, *i.e.,*

$$u_c(\theta(\varphi), t) = \sum_{i=1}^{n_{a_k}} a_i^k(p, t, \theta(\varphi)) u_c(\theta(\varphi), t-1) + \sum_{j=0}^{n_{b_k}} b_j^k(p, t, \theta(\varphi)) e_v(\varphi, t-j), \tag{6}$$

where $e_v(\varphi, t) = r_v(\varphi, t) - y(t)$ is the *fictitious* tracking error, computed by comparing the fictitious reference defined in (5) and the outputs available in the design phase.

The cost $J(\varphi, \theta)$ in (4) is chosen as

$$J(\varphi, \theta) = \frac{1}{T} \sum_{t=1}^{T} \left[e_v(\varphi, t)^2 + W_{\Delta u} \Delta u_c(\theta(\varphi), t)^2 + W_u (u(t) - u_c(\theta(\varphi), t))^2 \right], \tag{7a}$$

with $e_v(\varphi, t)$ being the fictitious tracking error, $u_c(\theta(\varphi), t)$ defined as in (6), and

$$\Delta u_c(\theta(\varphi), t) = u_c(\theta(\varphi), t) - u_c(\theta(\varphi), t-1), \quad t = 1, \ldots, T, \tag{7b}$$

where $u_c(\theta(\varphi), 0)$ is set to zero. Since $e_v(\varphi, t) = (M(\varphi, q)^{-1} - 1) y(t)$, the first term in the cost (7) penalizes the weighted difference between the reference model and an ideal unitary gain. Differently from [16], this allows us to account for the tracking performance of $M(\varphi, q)$, without requiring the user to fix the set-point to be tracked before-hand. Clearly, since the user selects both the structure of the reference model and the soft specifications on its parameters, the problem will then be solved by implicitly focusing on the performance for the set points of interest for the user. Instead, the second term in (7) penalizes large fluctuations in the reconstructed input sequence $\{u_c(\theta(\varphi), t)\}_{t=1}^{T}$ and the third one relates to how well the controller is capable of reconstructing the real input u. Indeed, the higher $W_{\Delta u}$ is with respect to W_u, the more the control input is forced to be smooth, at the price of a possible deterioration of the controller performance. Instead, the higher W_u is, the more aggressive the designed controller will be, compatibly with the available data and the fixed controller class $\mathcal{K}_p(\theta(\varphi))$. We stress that the last two terms in $J(\varphi, \theta)$ further allow us to account for the attainability of the reference model $M(\varphi, q)$, balancing out the effect of the first term in the cost. It is worth highlighting that, since the cost depends on the fictitious tracking error only, the proposed approach does not require one to close the loop during the training phase.

As previously underlined, the controller parameters θ are functions of the selected reference model. Due to this nested dependence on the parameters of the controller and the reference model, the minimization problem in (4) cannot be generally solved in closed form. Instead, we use *Bayesian optimization* (BO) [17] to explore the set of feasible reference models

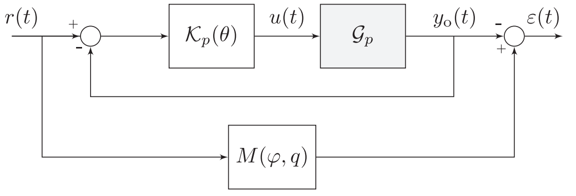


Fig. 2. Ideal matching scheme for direct LPV controller design, introduced in [8].

$M(\varphi, q)$, ultimately selecting the one leading to the best performance according to the cost in (7). This optimization procedure relies on the assumption that $J(\varphi, \theta(\varphi))$ can be modeled as a Gaussian process (GP), i.e., any finite set of observed values of $J(\varphi, \theta(\varphi))$ is seen as a joint Gaussian distribution with mean $\mu(\varphi)$ and covariances $\text{cov}(J(\varphi), J(\varphi')) = k(\varphi, \varphi')$, where $k \in \mathbb{R}$ is a covariance function. The Bayesian Optimization scheme exploits this joint distribution to make predictions on $J(\varphi)$ for some unobserved φ and find a minimum of the selected objective.

The overall procedure consists of the following steps (see also the flowchart in Fig. 3). Initially, N randomly chosen candidate reference models $\{M(\varphi^{(i)}, q)\}_{i=1}^{N}$ are introduced. For each tested reference model, an LPV controller in the class $\mathcal{K}_p(\theta(\varphi))$ indicated in (2) is constructed with the available dataset $\mathcal{D}_T$. With this controller, the cost (7) is computed and the set $\mathcal{B}_\varphi$ is constructed, which comprises the tested reference models and the corresponding costs, i.e.,

$$\mathcal{B}_\varphi = \left\{\varphi^{(i)}, J(\varphi^{(i)}, \theta(\varphi^{(i)}))\right\}. \tag{8}$$

The set $\mathcal{B}_\varphi$ is then used to compute preliminary values of the mean $m(\varphi)$ and variances $k_{ij}(\varphi_i, \varphi_j)$, $i, j \in [1, \cdots, N]$ of the joint distribution characterizing the GP of $J(\varphi, \theta(\varphi))$. This updated model is then exploited to select the next reference model to be tried out, which is chosen by maximizing a user-defined *acquisition function* $\alpha(\varphi|\mathcal{B}_\varphi)$. Accordingly, a new LPV controller is designed, the associated cost is evaluated, $\mathcal{B}_\varphi$ in (8) is augmented and $\mu(\varphi)$ and $\text{cov}(J(\varphi), J(\varphi'))$ characterizing the model of $J(\varphi, \theta(\varphi))$ are updated. This process is repeated until termination.

A key ingredient of the proposed flexible design strategy is the approach employed at each step to learn a new LPV controller. As already introduced, we rely on the method proposed in [8], due to its adaptability to different design scenarios. The exploited technique is based on the formulation of the following constrained optimization problem:

$$\min_{\theta, \varepsilon} \ \|\varepsilon\|_{\ell_2}^2 \tag{9a}$$

$$\text{s.t.} \quad \varepsilon(t) = M(\varphi, q)r(t) - y_o(t), \tag{9b}$$

$$\text{eq. } (1a) \text{ and eq.}(2) \text{ hold}, \quad \forall t = \{1, \ldots, T\}, \tag{9c}$$

where $\|\varepsilon\|_{\ell_2}^2$ denotes the ℓ_2-norm of ε, T is the length of the available dataset $\mathcal{D}_T$ and the constraints in (9c) are introduced to enforce the controller to belong to $\mathcal{K}_p(\theta)$ in (2) and to account for the dynamics of $\mathcal{G}_p$. The variable ε to be minimized is defined as in (9b), according to the matching scheme shown in Fig. 2. Note that, based on the BO scheme summarized in Fig. 3, this problem is solved whenever the reference model (and thus the parameter vector φ) is fixed.

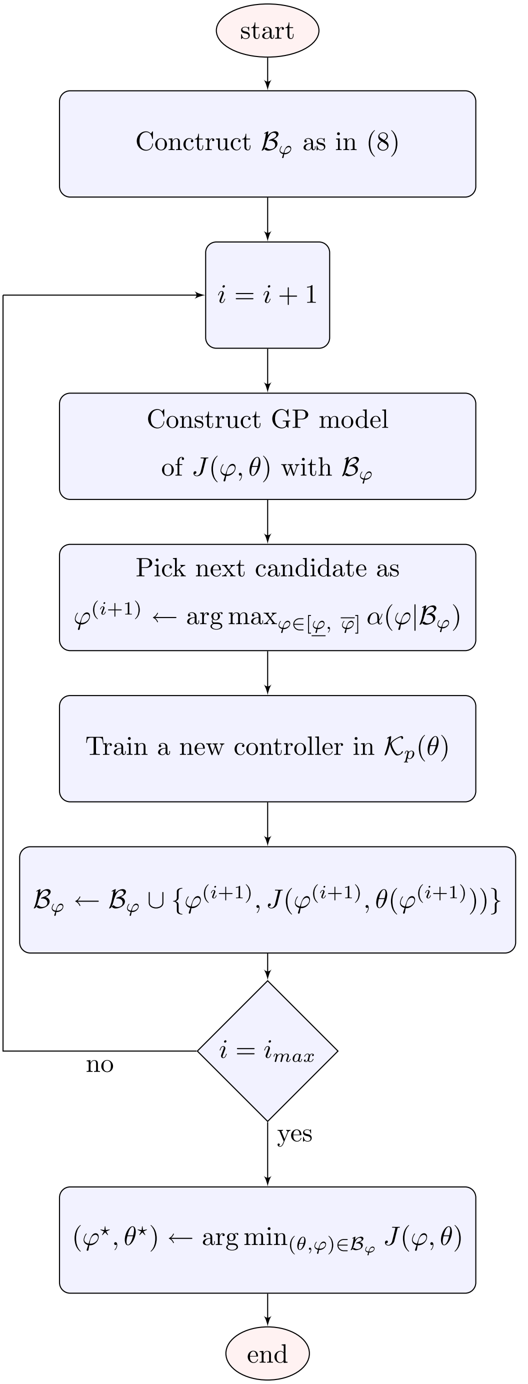


Fig. 3. Flowchart summarizing the main steps of the proposed design strategy.

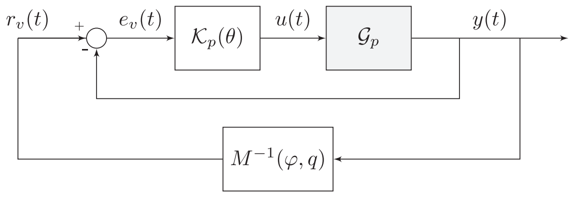


Fig. 4. Fictitious closed-loop scheme, exploited within all the phases of the proposed design procedure.

As it is, the problem in (9) cannot be solved since it requires the knowledge of plant dynamics. Furthermore, it imposes the user to fix the reference to be tracked, thus limiting the effectiveness of the resulting controller to unseen set points. These issues are overcome by (i) introducing a fictitious reference $\tilde{r}_v(\varphi, \varepsilon, t)$, that is defined via the constraint (9b) ; (ii) replacing $y_o(t)$ with the measured output, so that the *fictitious* closed-loop scheme used for training is the one reported in Fig. 4 and the constraint on the plant dynamics can be neglected.

Let $\tilde{r}_v(\varphi, \varepsilon, t) = r_v(\varphi, t) + M(\varphi, q)^{-1}\varepsilon(t)$, with $r_v(\varphi, t)$ defined as in (5). By exploiting this fictitious reference and rearranging the terms of (2), the optimization problem in (9) can be recast as:

$$\min_{\theta, \varepsilon} \; \|\varepsilon\|_{\ell_2}^2 \tag{10a}$$

$$\text{s.t. } B_K(p, t, \theta, q)M(\varphi, q)^{-1}\varepsilon(t) = A_K(p, t, \theta, q)u(t) - B_K(p, t, \theta, q)e_v(\varphi, t), \tag{10b}$$

where $e_v(\varphi, t) = r_v(\varphi, t) - y(t)$ is the *fictitious* tracking error, the constraint in (10b) has to hold for all $t \in \{1, \dots, T\}$ and $\mathcal{K}_p(\theta)$ be parametrized by

$$A_K(p, t, \theta, q) = 1 + \sum_{i=1}^{n_{a_k}} a_i^k(p, t, \theta)q^{-i}, \tag{11a}$$

$$B_K(p, t, \theta, q) = \sum_{j=0}^{n_{b_k}} b_j^k(p, t, \theta)q^{-j}. \tag{11b}$$

The reader is referred to [8] for further details on this derivation.

Note that the constraint in (10b) is bi-convex, due to the product between the polynomial $B_K(p, t, \theta, q)$ linearly depending on θ, and the optimization variable ε. As shown in [8], the form of this constraint is shaped by the availability of prior insights on a suitable controller class $\mathcal{K}_p(\theta)$ for $\mathcal{G}_p$, allowing us to design either parametric or non-parametric controllers, as summarized next.

3.1. Direct design of parametric LPV controllers

A rough information on the system might be sufficient to preselect the *structure* of the polynomials in (11), so that satisfactory performance is attained for reference models belonging to $\mathcal{M}(\varphi)$ in (3). In this scenario, the coefficients in (11) are defined as

$$a_i^k(p, t, \theta) = \sum_{l=1}^{m_i} \alpha_{i,l} f_{i,l}(p, t), \quad i = 1, \dots, n_{a_k} \tag{12a}$$

$$b_j^k(p,t,\theta)=\sum_{h=0}^{m_j}\beta_{j,h}g_{j,h}(p,t),\quad j=0,\dots,n_{b_k}, \tag{12b}$$

with $\{f_{i,l}(\cdot)\}_{l=1}^{m_i}$ and $\{g_{j,h}(\cdot)\}_{h=0}^{m_j}$ being nonlinear and (possibly) dynamic prefixed *basis functions* of the scheduling variable p, and where

$$\theta=\begin{bmatrix}\alpha_{1,1} & \cdots & \alpha_{n_{a_k},m_{n_{a_k}}} & \beta_{0,0} & \beta_{0,1} & \cdots & \beta_{n_{b_k},m_{n_{b_k}}}\end{bmatrix}' \tag{13}$$

collects all the unknown coefficients in (12). Since the basis functions are known, they can be precomputed with $\mathcal{P}_T$, leading to the following reformulation of the constraint in (10b):

$$B_K(p,t,\theta,q)M(\varphi,q)^{-1}\varepsilon(t)=u(t)-\phi'(\varphi,t)\theta, \tag{14}$$

with $\phi(\varphi,t)$ stacking $\{u(t-i)f_{i,l}(p,t)\}_{l=1}^{m_{n_{a_k}}}$ and $\{e_v(\varphi,t-j)g_{j,h}(p,t)\}_{h=0}^{m_j}$, for $i=1,\dots,n_{a_k}$ and $j=1,\dots,n_{b_k}$.

To deal with measurement noise and remove the bi-convex constraint in (14), we rely on a regularized *instrumental variable* (IV) scheme. The design problem to be solved thus becomes

$$\min_{\theta}\left\|\sum_{t=1}^{T}\zeta(t)\big(\phi'(\varphi,t)\theta-u(t)\big)\right\|_2^2+\lambda\|\theta\|^2, \tag{15}$$

where $\lambda>0$ is a regularization parameter introduced to better condition the problem. The *instrument* $\zeta(t)$ is a vector with the same dimension of $\phi(\varphi,t)$ but uncorrelated with the output noise[1]

Let $Z_T(\varphi)=\sum_{t=1}^{T}\zeta(t)\phi(\varphi,t)'$. Accordingly, the closed-form solution for problem (15) is given by

$$\theta^\star(\varphi)=\big(Z_T'(\varphi)Z_T(\varphi)+\lambda I\big)^{-1}\left[Z_T'(\varphi)\sum_{t=1}^{T}\zeta(t)u(t)\right] \tag{16}$$

3.2. Direct design of non-parametric LPV controllers

Since the reference model is flexible and considering that prior knowledge on $\mathcal{G}_p$ may not available, it is quite likely that the structure of a suitable controller cannot be easily selected a priori.

In this scenario, the coefficients in (11) can be defined as [8]:

$$a_i^k(p,t,\theta)=\theta_i'\psi_i(p,t),\quad b_j^k(p,t,\theta)=\theta_{d_j}'\psi_{d_j}(p,t), \tag{17}$$

with $d_j=n_{a_k}+1+j$, $\theta_i\in\mathbb{R}^{n_H}$. The functions $\psi_i:\mathbb{P}\to\mathbb{R}^{n_H}$, for $i=1,\dots,n_{a_k}+n_{b_k}+1$, are now *unknown* nonlinear maps that relate the scheduling parameter space to the feature space $\mathbb{R}^{n_H}$, whose dimension n_H is also supposed to be unknown. Let the regressor $x_v(\varphi,t)$ be defined as

$$x_v(\varphi,t)=\begin{bmatrix}-u(t-1) & \dots & -u(t-n_{a_k}) & e_v(\varphi,t) & \dots & e_v(\varphi,t-n_{b_k})\end{bmatrix}'. \tag{18}$$

[1] A possible approach to construct $\zeta(t)$ is to build a new version of $\phi(\varphi,t)$, based on data collected over a second open-loop experiment carried out with the same input sequence $\mathcal{U}_T$. In this case the instrument is a function of φ.

Accordingly, the constraint in (10b) becomes

$$u(t) = \sum_{i=1}^{n_f} \theta_i' \psi_i(p, t) x_{v,i}(\varphi, t) + \varepsilon_u(t), \tag{19}$$

where $x_{v,i}(\varphi, t)$ denotes the i-th component of $x_v(\varphi, t)$, $n_f = n_{a_k} + n_{b_k} + 1$, and $\varepsilon_u(t)$ is a new variable to be optimized, that replaces the bi-convex term $B_k(p, t, \theta, q)M^{-1}(\varphi, q)\varepsilon(t)$.

By exploiting the LS-SVM framework [18], problem (10) can then be cast as

$$\begin{aligned} &\min_{\theta_i, \varepsilon_u} \ \frac{1}{2} \sum_{i=1}^{n_f} \left[\theta_i' \theta_i + \frac{\gamma}{T^2} \left\| \sum_{t=1}^{T} \zeta_i(t) \varepsilon_u(t) \right\|_2^2 \right] \\ &\text{s.t. } \varepsilon_u(t) = u(t) - \sum_{i=1}^{n_f} \theta_i' \psi_i(p, t) x_{v,i}(\varphi, t), \ \ t = 1, \ldots, T. \end{aligned} \tag{20}$$

Note that the problem to be solved is regularized to prevent over-fitting, and the regularization strength controlled by the tunable parameter $\gamma > 0$. This formulation also exploits an instrumental variable scheme, with instruments $z_i(t)$. Each instrument is chosen as $z_i(t) = \psi(p, t)\tilde{x}_i(\varphi, t)$, where $\tilde{x}_i(\varphi, t)$ is a new realization of the regressor defined in (18). For additional details, the reader is referred to [8].

Due to the convexity of the cost and the nature of the constraints, the global optimum for the problem in (20) can be found by imposing the *Karush-Kuhn-Tucker* (KKT) conditions. This requires the introduction of an additional set of variables to be retrieved, namely the *Lagrange multipliers* $\{\delta(t)\}_{t=1}^{T}$ associated with the constraints, that are needed to find the optimal LPV controller. As detailed in [8], the Lagrange multipliers can be computed by exploiting the so-called *kernel trick* [19], where the products of the unknown functions ψ_i are replaced by some known kernel functions $\{\kappa_i(p, t, \tau)\}_{i=1}^{n_f}$, that embed the structure of the controller class $\mathcal{K}_p(\theta)$.

Let $\delta \in \mathbb{R}^T$ be the vector stacking all the Lagrange multipliers and let $[\Omega_i]_{t,\tau} = \kappa_i(p, t, \tau)$, it can be proven (see [8]) that the following holds:

$$\delta = R_D(\Omega_i)^{-1} \frac{1}{T^2} \sum_{i=1}^{n_f} \tilde{X}_i \Omega_i \tilde{X}_i U, \tag{21a}$$

where $U \in \mathbb{R}^T$ is the vector stacking the collected inputs, $R_D(\Omega_i)$ is given by

$$R_D(\Omega_i) = \gamma^{-1} I + \frac{1}{T^2} \sum_{i=1}^{n_f} \tilde{X}_i \Omega_i \tilde{X}_i \sum_{j=1}^{n_f} X_j \Omega_i X_j, \tag{21b}$$

and X_i and $\tilde{X}_i$ are diagonal matrices, such that it holds that $[X_i]_{\tau\tau} = x_i(\varphi, \tau)$ and $[\tilde{X}_i]_{\tau\tau} = \tilde{x}_i(\varphi, \tau)$, for $\tau = 1, \ldots, T$.

Once the Lagrange multipliers are retrieved, the coefficients in (11) can finally be computed as

$$a_i^k(p, t, \theta) = \sum_{\tau=1}^{T} \kappa_i(p, \tau, t) x_i(\varphi, \tau) \delta(\tau), \tag{22a}$$

$$b_j^k(p, t, \theta) = \sum_{\tau=1}^{T} \kappa_{d_j}(p, \tau, t) x_{d_j}(\varphi, \tau) \delta(\tau), \tag{22b}$$

with $d_j = n_{a_k} + 1 + j$, which concludes the controller design phase.

Remark 1 (On the choice of kernels)**.** The kernel functions to be selected are known to shape the performance of the resulting LPV controller. A proper choice of the kernels is thus crucial to attain good closed-loop performance. Although this choice heavily depends on the problem at hand, *Radial Basis Function* (RBF) kernels can be used, with

$$\kappa_i(p, \tau, t) = \exp\left(-\frac{\|p(\tau) - p(t)\|_2^2}{\sigma_i^2}\right), \tag{23}$$

since they effectively represent a wide range of smooth functions. Here, σ_i is an hyper-parameter to be tuned, that characterizes the width of the RBF.

Remark 2. As we exploit the strategy proposed in [8] to directly design the LPV controller from data, the user has to tune the regularization parameter λ in (16) or γ in (20) and the kernel widths $\{\sigma_i\}_{i=1}^{n_f}$ in (23), depending on the chosen design framework. Within our setting, these design hyper-parameters can be selected by adding an additional optimization layer within the proposed Bayesian Optimization framework, thus reducing the burden on the user side at the price of a longer design procedure.

4. Examples

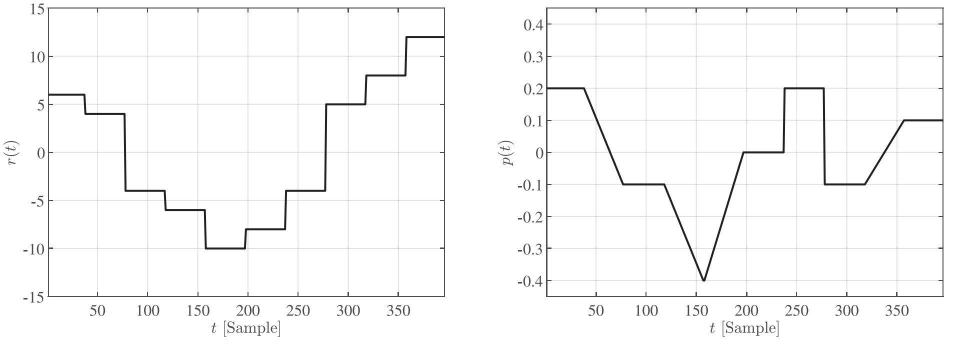
To assess the benefits of the proposed design strategy, we address the same problems already considered in [8]. In the first example, we design an LPV controller by relying on a flexible second-order reference model, which is selected according to constraints imposed on the acceptable settling time and overshoot to step-like references. Then, we consider a rather realistic case study, designing an LPV controller for a voltage-controlled DC motor. In this last example we perform a detailed sensitivity analysis to the design parameter, that might be useful as a tuning guideline even for different applications.

4.1. Performance assessment

In both examples to quantitatively assess the attained closed-loop performance, we introduce the following indicators:

$$\begin{aligned}
\mathrm{RMSE_M}(\theta, \varphi) &= \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y(\theta(\varphi), t) - y_d(\varphi, t))^2},\\
\mathrm{RMSE_Y}(\theta, \varphi) &= \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y(\theta(\varphi), t) - r(t))^2}.
\end{aligned} \tag{24}$$

The first index, namely $\mathrm{RMSE_M}(\theta, \varphi)$, relates to how well the closed-loop response matches the desired behavior, thus measuring the adherence to design constraints. Instead, $\mathrm{RMSE_Y}(\theta, \varphi)$ quantifies the ability of the closed-loop system to track a user-defined reference $r(t)$. We stress that these quality indexes can be easily computed since we work within a simulation environment, which allows us to run closed-loop experiments without threatening the safety of the actual process.



(a) (b)

Fig. 5. The reference $r(t)$ to be tracked is displayed in (a), while (b) shows the imposed to scheduling sequence $p(t)$. These signals correspond to the ones considered in [8].

4.2. Numerical example

Consider the following SISO LPV plant

$$\begin{aligned} x_G(t+1) &= p(t)x_G(t) + u(t) \\ y(t) &= x_G(t) + w(t), \end{aligned} \tag{25}$$

with $-0.2 \leq p(t) \leq 0.2$. Assume that we aim at attaining a desired closed-loop behavior described through the following second-order LTI reference model

$$M(\varphi, q) = \frac{q^{-2}\big(1 - e^{(\varphi_1 + j\varphi_2)}\big)\big(1 - e^{(\varphi_1 - j\varphi_2)}\big)}{\big(1 - q^{-1}e^{(\varphi_1 + j\varphi_2)}\big)\big(1 - q^{-1}e^{(\varphi_1 - j\varphi_2)}\big)}. \tag{26}$$

As in [12], let us specify the range of feasible models by imposing bounds on the maximum overshoot $s_{\max}$ and the settling time T_a to step-like references, with $s_{\max} \in [10^{-6},\ 1]$ and $T_a \in [2,\ 10]$ s. To simultaneously obtain an optimal second-order reference model $M(\varphi^\star, q)$ and LPV controller, without specifying the structure of its parameters, we apply the design procedure described in Section 3.2. To this end, a set $\mathcal{D}_T$ of open-loop measurements of length $T = 1000$ is collected using zero-mean input $u(t)$ with a uniform distribution. The scheduling signal during these experiments is selected as $p(t) = 0.4 \sin(2\pi \cdot 0.03t)$, while the output measurements are subject to measurement noise $w(t) \sim \mathcal{N}(0, 0.2^2)$, resulting in a *Signal-to-Noise Ratio* (SNR) of 9.8 dB. In this test, the regularization parameter is fixed at $\gamma = 77844$ and the hyper-parameter of the kernel is chosen as $\sigma = 2.9$. The Bayesian Optimization procedure described in Section 3 is run for 30 times, with $W_{\Delta u} = 0.02$, $W_u = 10$.

The performance attained with the designed controller is assessed with respect to the same reference and scheduling sequence considered in [8], that are both depicted in Fig. 5.

As shown in Fig. 6, the considered step-like set point is tracked. This indicates that the reference model automatically chosen by the proposed BO-based approach leads to an achievable closed-loop behavior, as proven by the value of RMSE_M reported in Table 1. Note that the optimal settling time and overshoot shown in the Table 1 lie within the imposed constraints.

We further compare the results achieved with the optimized reference model with the ones attained with a second-order $M(\tilde{\varphi}, q)$, with $\tilde{\varphi}$ fixed before-hand. In selecting the latter

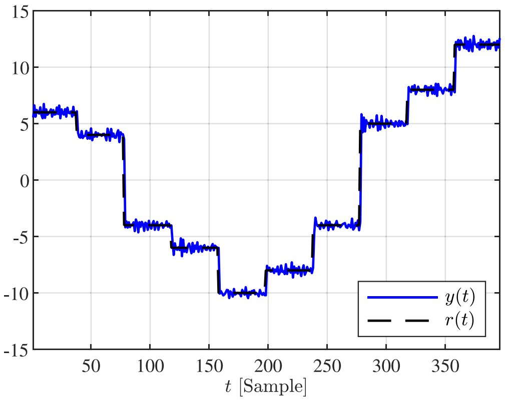


Fig. 6. Attained set-point tracking performance in closed-loop.

Table 1
Numerical example: fixed vs flexible reference model.

	$M(\tilde{\varphi}, q)$	$M(\varphi^{\star}, q)$
T_a [s]	10.00	2.03
$s_{\max}$	0.100	0.58
RMSE_{M}	0.19	0.22
RMSE_{y}	0.92	0.78

reference model, conservative desired performance have been considered, so as to represent a realistic situation in which the user has no information on the plant to be controlled and, thus, on achievable closed-loop performance. The quantitative indexes defined in (24) attained with the fixed and flexible reference model are shown in Table 1, highlighting that the proposed flexible design method is effective at trading-off between set point tracking performance and achievability of the desired closed-loop behavior.

4.3. A simulation case study: The DC motor

Consider now the voltage-controlled DC motor considered in [8], that features an inhomogeneous mass distribution caused by a mass placed on the disc connected to the rotor (see Fig. 7). This servo-positioning system is described by the continuous-time dynamics:

$$\dot{x}(t) = \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{K}{J} \\ 0 & -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ \frac{Mgl}{J} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{\sin(\theta(\tau))}{\theta(\tau)} \right) x(t) + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} V(t). \tag{27a}$$

The state and the output of the system are respectively given by

$$x(t) = \left[\theta(t),\ \omega(t),\ I(t)\right]^T, \qquad y(t) = \theta(t), \tag{27b}$$

where $\theta(t)$ [rad] is the motor angular position, $\omega(t)$ [rad/s] is its angular velocity and $I(t)$ [mA] denotes its current. The controlled input to the system is the voltage $V(t)$ [V] and the remaining constants characterizing the DC motor are shown in Table 2. It is worth

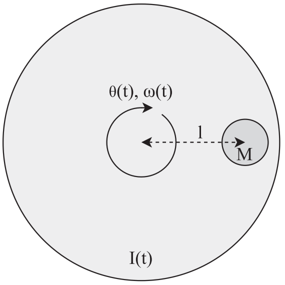


Fig. 7. Schematic representation of the unbalanced disk in the example problem.

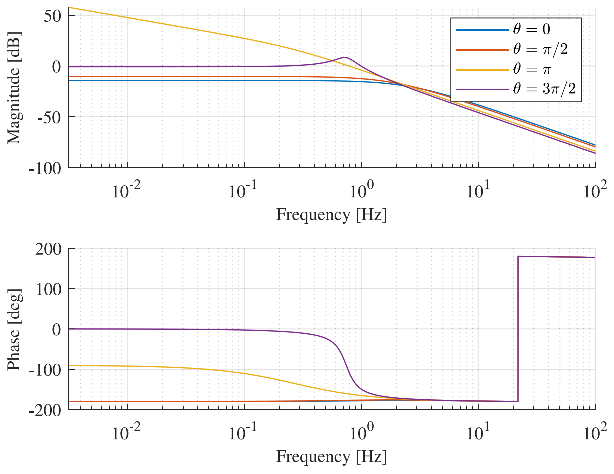


Fig. 8. Bode plots of the servo system for some fixed values of θ.

Table 2
Parameters of the DC motor- [20].

Symbol	Description	Value
R	Motor resistance	9.5 [Ω]
L	Motor inductance	$0.84 \cdot 10^{-3}$ [H]
K	Motor torque constant	$53.6 \cdot 10^{-3}$ [N m/A]
J	Complete disk inertia	$2.2 \cdot 10^{-4}$ [N m^2]
b	Friction coefficient	$6.6 \cdot 10^{-5}$ [N m s/rad]
M	Extra mass	0.07 [kg]
l	Mass distance from the center	0.042 [m]

stressing that the considered system is quasi-LPV with $p(t) = \theta(t)$. Note that, the plant dynamics in (27a) fits the definition of $\mathcal{G}_p$ provided in (1a) once converted in discrete-time and in input/output form. For some fixed values of θ, the frequency response of the dynamics is shown in Fig. 8.

4.3.1. Setup and goal

Our objective is to achieve an LTI behavior in closed-loop for the DC motor. For ease of comparison with the results obtained in [8], we focus on first order LTI flexible reference models:

$$\mathcal{M}(\varphi): \; y_d(\varphi, t) = \underbrace{\frac{q^{-1}(1-e^{\varphi})}{1-q^{-1}e^{\varphi}}}_{M(\varphi,q)} r(t), \tag{28}$$

and we impose bounds on the desired settling time for step-like responses. Specifically, we constrain the parameter φ within the following set

$$\varphi \in [-0.5, \; -0.005], \tag{29}$$

so as to enforce the settling time of the closed-loop step response to be comprised within [0.1,10] s. Note that, we exactly retrieve the reference model considered in [8] by imposing $\varphi = \log(0.99) \approx -0.01$.

To assess the effectiveness of the proposed strategy within the two possible design scenarios that it handles, both a *parametric* LPV controller and a *non-parametric* one are synthesized by using the angular position $\theta(t)$ as scheduling variable and by exploiting an integrator, so as to achieve zero steady-state error, as in [8]. Accordingly, the controller class is modified as:

$$\mathcal{K}_p(\theta): u(t) = \sum_{i=1}^{n_{a_k}} a_i^k(p,t,\theta)u(t-i) + \sum_{j=0}^{n_{b_k}} b_j^k(p,t,\theta)e_{int}(t-j), \tag{30a}$$

with

$$e_{\text{int}}(t) = e_{\text{int}}(t-1) + (r(t) - y(t)). \tag{30b}$$

When designing the *parametric* LPV controller, we impose $n_{a_k} = 4$, $n_{b_k} = 5$ and select the following basis functions:

$$f_{i,l}(p,t) = \frac{\sin(p(t-l))}{p(t-l)}, \;\; l = 1, \dots, 4, \forall i \in \{1, \dots, n_{a_k}\},$$
$$g_{j,h}(p,t) = \frac{\sin(p(t-h))}{p(t-h)}, \;\; h = 1, \dots, 4, \forall j \in \{0, 1, \dots, n_{b_k}\},$$

by keeping a linear term on the integral of the tracking error, so that the dynamics of the controller is given by

$$u(t) = \sum_{i=1}^{n_{a_k}} a_i^k(p,t,\theta)u(t-i) + \\ + \sum_{j=0}^{n_{b_k}} b_j^k(p,t,\theta)e_{int}(t-j) + \sum_{j=0}^{n_{b_k}} \tilde{b}_j^k e_{int}(t-j). \tag{31}$$

This structure is build upon our prior knowledge of the nonlinearity characterizing the dynamics in (27a). Even though this prior allows us to select a suitable controller structure, its choice is still rather challenging and the chosen structure is yet not optimal. As such, we do not expect the parametric LPV controller to outperform the non-parametric one presented in [8], despite the flexibility of the reference model.

When we exploit the *non-parametric* approach, we consider the same structure exploited in [8], namely we set $n_{a_k} = n_{b_k} = 4$, and we impose

$$a_i^k(p,t) = a_i^k(\Pi(t)), \;\; \forall i = 1, \dots, n_{a_k}, \tag{32a}$$

Table 3
Flexible design: parametric vs non-parametric approach.

	Parametric $\mathcal{K}_p(\theta)$	Non-parametric $\mathcal{K}_p(\theta)$
$W_{\Delta u}$	0.06615	0.0215
W_u	1098.5	46416
$\exp(\varphi^\star)$	0.9981	0.9768
$\lambda^\star$	$1.8417{\cdot}10^{-3}$	-
$\gamma^\star$	-	$1.1150{\cdot}10^{6}$
$\mathrm{RMSE_y}$	0.4014	0.2481
$\mathrm{RMSE_M}$	0.1486	0.0535

$$b_j^k(p,t) = b_j^k(\Pi(t)),\ \ \forall j = 0, 1, \ldots, n_{b_k}, \tag{32b}$$

with

$$\Pi(t) = \begin{bmatrix} p(t-1) & p(t-2) & p(t-3) & p(t-4) \end{bmatrix}'. \tag{32c}$$

We stress that the additional flexibility guaranteed by a non-parametric controller structure, juxtaposed with the non-optimality of the parametric controller, is expected to lead to improved performance of the non-parametric controller with respect to its parametric counterpart.

To design the controller, we always rely on an experimental configuration similar to the one used in [8]. A set of open-loop measurements is collected by feeding the DC motor with a Gaussian distributed white noise sequence, with mean equal to 16 V. Prior to be supplied to the system, this signal is filtered via a first-order low-pass filter, with cutoff frequency of 1.6 Hz. The measured angular position is affected by zero-mean white noise with Gaussian distribution, whose variance yields a *Signal-to-Noise Ratio* (SNR) equal to 43 dB, which is rather realistic for the application at hand. All data are acquired using an ideal *zero-order-hold* (ZOH) scheme with sampling time 0.01 s, with an additional filter acting on the output to avoid aliasing. A set $\mathcal{D}_T$ of open-loop measurements of length $T = 3000$ is used when designing the parametric LPV controller, while it is reduced to $T = 1500$ in the non-parametric case, due to the increased computational complexity. The same input sequence $\mathcal{U}_T$ is exploited to carry out an additional experiment, to compute the instrument.

The BO-based approach proposed in Section 3 is carried out for a maximum number of iterations $i_{\max}$ equal to 50, by using the *Expected Improvement* (EI) acquisition function to pick the next reference model to be tested and Matérn kernels to update the model of the cost function (7), see [21,22]. The regularization parameters λ in (16) and γ in (20) are treated as additional design variables and optimized through Bayesian optimization, along with the parameters of the controller θ and the reference model. On the other hand, $\{\sigma_i\}_{i=1}^{n_f}$ in (23) are all fixed to 2.4, so as to limit the computational burden. It is worth remarking that the kernel width can be incorporated as a design variable as well, if there are no strict limits on the computational power.

4.3.2. Results

The weights exploited to train the parametric and non-parametric LPV controller are reported in Table 3, along with the resulting optimal reference model and regularization parameters. As indicated there, $\varphi^\star$ obtained when training a non-parametric LPV controller

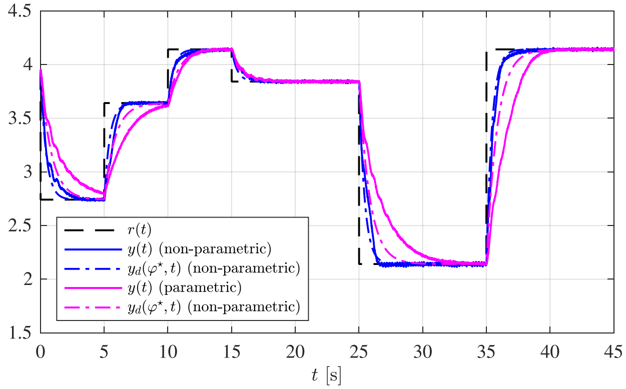


Fig. 9. Closed-loop response using parametric and non-parametric controllers, obtained by for their corresponding optimal reference model $M(\varphi^\star)$.

Table 4
Nonparametric controller: flexible vs fixed [8] reference model.

	flexible $\mathcal{M}(\varphi)$	Nominal $\mathcal{M}(\varphi)$
$\exp(\varphi^\star)$	0.9768	0.9900
$\gamma^\star$	$1.1150\cdot10^6$	64163
$\mathrm{RMSE_y}$	0.2481	0.3533
$\mathrm{RMSE_M}$	0.0535	0.0493

yields a significantly more reactive closed-loop system than the one attained with a parametric controller, as confirmed by the results shown in Fig. 9. These responses and the $\mathrm{RMSE_M}$ reported in Table 3 further indicate that the use of a non-parametric design strategy leads to a closed-loop response that is more adherent to the desired one, even though the latter is more demanding. At the same time, these results prove the sub-optimality of the chosen parametric controller, whose structure indeed requires us to select a less performing reference model for its behavior to be matched in closed-loop. This highlights once more that structure selection might be a rather challenging task, despite the priors available on the plant to be controlled. We stress that the improvement in the response is attained at the price of an increased computational time required to learn the non-parametric controller. Specifically, training is 100 times faster in the parametric case, where $i_{\max}$ iterations of Bayesian optimization take around 1 minute[2] A comparison between the performance of the non-parametric LPV controller obtained with our training procedure with the one attained by fixing the reference model according to [8] is then shown in Fig. 10. It is clear that the optimization of the cost $J(\theta, \varphi)$ in (7) leads to a reference model that is achievable, but yet characterized by a higher bandwidth than the one selected (more conservatively) a-priori. This is further indicated by the results reported in Table 4. It has to be pointed out that this result is also linked to the optimization of γ, which is now more than one order of magnitude higher than the one cho-

[2] The parametric controller has been trained and tested on an i7 3.40-GHz Intel core processor with 16 GB of RAM, while in the non-parametric case an i7 2.50-GHz Intel core processor with 8 GB of RAM is used. In both cases, the system runs MATLAB R2019b.

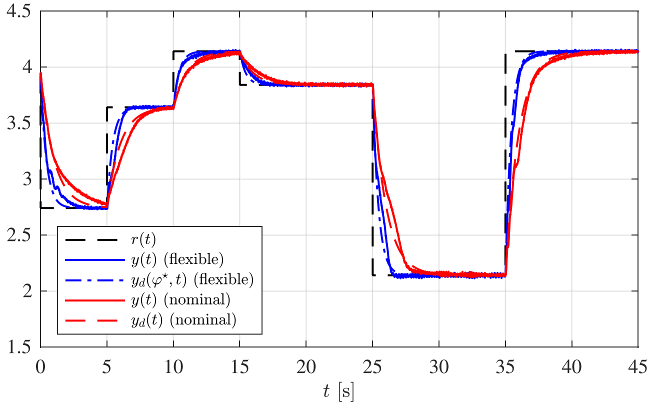


Fig. 10. Closed-loop response using non-parametric controllers. The result obtained from optimizing for the optimal reference model $M(\varphi^{\star})$ is compared to the result obtained from using the nominal reference model from [8].

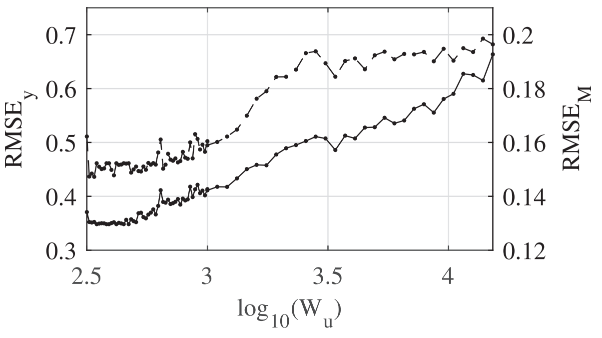


Fig. 11. Parametric controller performance vs W_u: average RMSE_y (left axis, solid line) and RMSE_M (right axis, dashed line) computed over 40 different values of $W_{\Delta u} \in [0.01,\ 1]$ at each tested value of W_u.

sen through cross-validation in [8] (see Table 4). By incorporating this parameter within our design variables, we account for the sensitivity of the closed-loop performance to the choice of γ, which indeed plays an important role as shown in the following sensitivity analysis.

4.3.3. Sensitivity analysis

Although the optimization of the cost (7) is performed with open-loop data only, one has no guarantees that its minimizer leads to the optimal performance according to (24). By focusing now on the parametric case only, we investigate how the choice of the weights W_u and $W_{\Delta u}$ influences the final closed-loop behavior in terms of $\text{RMSE}_\text{M}(\theta, \varphi)$ and $\text{RMSE}_\text{y}(\theta, \varphi)$.

A comparison between these two quality indexes and the value of W_u is reported in Fig. 11. Accordingly, the lower the value of this weight is, the better performance is attained in closed-loop both in terms of set-point tracking and model reference mismatch. We stress that the values for W_u considered in Fig. 11 are still (at least) two orders of magnitude higher than the unitary weight attributed to the virtual tracking error, suggesting that the flexible reference model should be trained by looking mainly at input mismatches, at least in this case study. Instead, by investigating the effect of different choices of $W_{\Delta u}$ on the resulting closed-loop performance, no actual trend can be detected. This result can be related to the fact that $W_{\Delta u}$

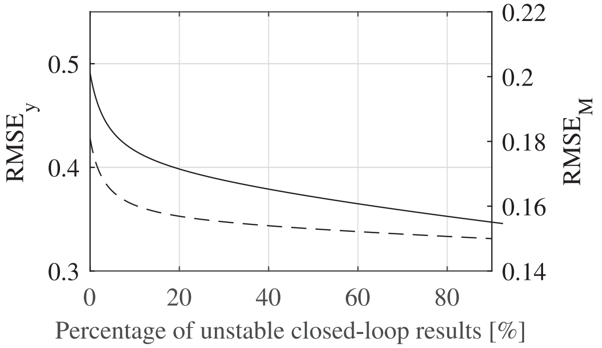


Fig. 12. Parametric controller vs percentage of observed unstable closed-loop responses: average RMSE_y (left axis, solid) and RMSE_M (right axis, dashed).

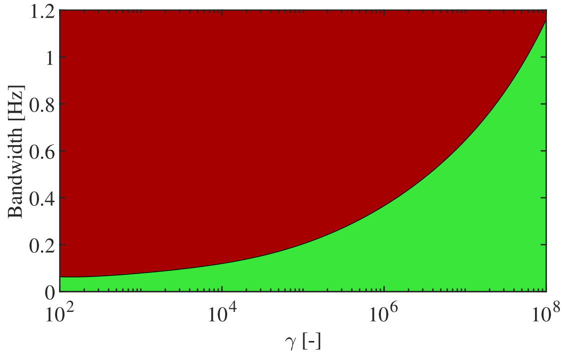


Fig. 13. Nonparametric controller: achievable closed-loop bandwidth (green area) as a function of the regularization parameter γ. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

is not exploited to select the regularization parameter $\lambda^\star$, that is instead optimized along with the other design parameters. This seems to suggest that, if this weight is not used to shape the regularization strength, the related term might be left out of the cost without jeopardizing the closed-loop performance.

In the results reported in Fig. 11, we have excluded a small number of outliers, that represent the cases where the obtained controller led to an unstable closed-loop response even for a finite cost $J(\varphi^\star, \theta^\star)$. This unwanted behavior is mainly observed for smaller values of W_u, as can be deduced from the results reported in Fig. 12, thus indicating that the closed-loop system is prone to be unstable when W_u is chosen so as to minimize the considered performance indexes (see their definition in (24)).

This suggests that it might be wiser to consider large values of W_u at first and then proceed by carefully lowering this weight to improve the closed-loop performance.

Let us now consider the non-parametric case. Instead of focusing on the weights in the cost (7), here we study how different choices of γ might affect the resulting closed-loop performance. Indeed, it is well known that the value of γ influences the result of LS-SVM [18].

To this end, the method proposed in Section 3 has been run by fixing different values for γ before-hand. The bandwidths of the corresponding optimized reference models are compared in Fig. 13, showing that the lower γ is, the more conservative the automatically chosen reference model will be. This yields the conclusion that the lower bound on γ should be sufficiently high if one seeks for an aggressive controller. At the same time, we have found that the closed-loop becomes unstable for values of γ higher than approximately 10^7, even for finite values of J in (7), which is likely to imply that the resulting reference model is too demanding and thus not achievable. This implies that one cannot push γ to be excessively high, thus limiting the upper-bound that can be imposed when optimizing it.

5. Conclusions

In this paper, an approach for direct design of LPV controllers with flexible reference models is presented. By treating the reference model as an *hyper-parameter*, the developed approach requires the user to define a range of admissible closed-loop responses, rather than fixing the desired closed-loop behavior before-hand. This leads to a more user-friendly design strategy, that allows for the imposition of some specifications on the closed-loop response, while not excessively constraining the design before-hand.

From a methodological perspective, future research will be devoted to the generalization of the approach to non-minimum phase plants and to alternative classes of controllers. Moreover, some effort will be devoted towards the definition and the assessment of alternative objectives to be optimized to concurrently select the flexible reference model and design a controller from data. This will allow us to look at alternative closed-loop features than tracking performance, *e.g.,* robustness against mismatches, thus enabling us to provide useful practical guidelines on the objective to be considered depending on the needs of the user. We also aim at testing the approach on real-world case studies, for a more complete experimental assessment of its effectiveness.

Declaration of Competing Interest

None.

References

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

[2] H. Hjalmarsson, M. Gevers, S. Gunnarsson, O. Lequin, Iterative feedback tuning: theory and applications, IEEE Control Syst. Mag. 18 (4) (1998) 26–41.

[3] O. Lequin, M. Gevers, M. Mossberg, E. Bosmans, L. Triest, Iterative feedback tuning of PID parameters: comparison with classical tuning rules, Control Eng. Pract. 11 (9) (2003) 1023– 1033. Special Section on Algorithms and Applications of Iterative Feedback Tuning

[4] S. Formentin, A. Karimi, Enhancing statistical performance of data-driven controller tuning via l2-regularization, Automatica 50 (5) (2014) 1514– 1520.

[5] 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.

[6] J. Berberich, J. Köhler, M. Müller, F. Allgöwer, Data-driven model predictive control with stability and robustness guarantees, IEEE Transactions on Automatic Control 66 (2021) 1702–1717.

[7] J. Coulson, J. Lygeros, F. Dörfler, Data-enabled predictive control: In the shallows of the deepc, in: 2019 18th European Control Conference (ECC), 2019, pp. 307–312.

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

[9] S. Formentin, S.M. Savaresi, Virtual reference feedback tuning for linear parameter-varying systems, IFAC Proceed. 44 (1) (2011) 10219– 10224. 18th IFAC World Congress.

[10] V. Breschi, S. Formentin, Direct data-driven design of switching controllers, Int. J. Robust Nonlinear Control (2019).

[11] M. Tanaskovic, L. Fagiano, C. Novara, M. Morari, Data-driven control of nonlinear systems: an on-line direct approach, Automatica 75 (2017) 1– 10.

[12] V. Breschi, S. Formentin, Proper closed-loop specifications for data-driven model-reference control, in: 24th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2020), 2020.

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

[14] P. Kergus, M. Olivi, C. Poussot-Vassal, F. Demourant, From reference model selection to controller validation: application to Loewner data-driven control, IEEE Control Syst. Lett. 3 (4) (2019) 1008–1013.

[15] A. Lecchini, M. Gevers, On iterative feedback tuning for non-minimum phase plants, in: Proceedings of the 41st IEEE Conference on Decision and Control, 2002., 4, 2002, pp. 4658– 4663.

[16] D. Selvi, D. Piga, A. Bemporad, Towards direct data-driven model-free design of optimal controllers, in: 2018 European Control Conference (ECC), 2018, pp. 2836–2841.

[17] E. Brochu, V.M. Cora, N. de Freitas, A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning (2010), arXiv:1012.2599.

[18] V. Laurain, R. Tóth, W.-X. Zheng, M. Gilson, Nonparametric identification of LPV models under general noise conditions: an LS-SVM based approach, IFAC Proceed. 45 (16) (2012) 1761– 1766. 16th IFAC Symposium on System Identification

[19] V.N. Vapnik, Statistical Learning Theory, Wiley-Interscience, 1998.

[20] B. Kulcsár, J. Dong, J.-W. van Wingerden, M. Verhaegen, LPV Subspace identification of a DC motor with unbalanced disc, IFAC Proceed. 42 (10) (2009) 856–861.

[21] R.M. Neal, Bayesian Learning for Neural Networks, Springer New York, 2012.

[22] The MathWorks Inc., MATLAB Statistics and Machine Learning Toolbox, The MathWorks Inc., Natick, MA, USA, 2019.