On data-driven design of LPV controllers with flexible reference models
Valentina Breschi, Max van Meer, Tom Oomen, and Simone Formentin
In 4th IFAC Workshop on Linear Parameter Varying Systems, Milan, Italy, 2021

On data-driven design of LPV controllers with flexible reference models ⋆

V. Breschi *, **M. van Meer** **, **T. Oomen** **, **S. Formentin** *

* *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, via G. Ponzio 34/5, 20133 Milano, Italy (e-mail: valentina.breschi@polimi.it)*
** *Department of Mechanical Engineering, Building 15, Gemini-Zuid, Eindhoven University of Technology, Eindhoven, The Netherlands*

Abstract: Many data-driven control design methods require the *a-priori* selection of a reference model to be tracked. In case of limited priors on the plant, such a blind choice might ultimately compromise the overall performance. In this work, we propose a nested strategy for the direct design of Linear Parameter Varying (LPV) controllers from data, in which the reference model is treated as a *hyperparameter* to be tuned. The proposed approach allows one to jointly optimize the reference model and learn an LPV controller, solely based on *soft* specifications on the desired closed-loop. The effectiveness of the proposed technique is assessed on a benchmark case study, with the obtained results showing its potential advantages over a state-of-the-art method.

Keywords: Data driven control, Non-parametric Methods

1. INTRODUCTION

Learning-based control solutions are becoming increasingly appealing, due to the widespread availability of data and the concurrent growth in complexity of the systems to be controlled. Differently from more traditional two-stage strategies, several approaches have been proposed over the last decades to directly design controllers from data, so as to avoid the usually expensive and time consuming initial modelling phase. Examples of direct design techniques range from the recent works on data-enabled predictive control, *e.g.,* (Berberich et al., 2021), to more traditional strategies, such as the *Virtual Reference Tuning* (VRFT) method (Campi et al., 2002) and the *Iterative Feedback Tuning* (IFT) approach (Hjalmarsson et al., 1998). Thanks to the latest advances in nonlinear identification, state-of-the-art direct control strategies have also been extended to handle more complex scenarios, *e.g.,* see (Formentin et al., 2016). Nonetheless, a key requirement in more traditional approaches to direct control design is the definition of a reference model, that describes the desired closed-loop behavior. Usually, this choice has to be performed beforehand and it might be rather challenging if one has little to no prior knowledge on the plant. Indeed, one might end up requiring excessively demanding performance, that would jeopardize the whole design procedure. On the other hand, conservatively selecting low demanding reference models might result in much poorer performance than the ones actually achievable.

With a focus on LPV control structures, in this work we aim at overcoming this major limitation by reconsidering the role of reference models on the same line as (Breschi and Formentin, 2020; Campestrini et al., 2011; Lecchini and Gevers, 2002; Selvi et al., 2018). While its structure is still fixed, here the parameters of the reference model are treated as tunable *hyperparameters*. This *change of perspective* allows us to introduce a new degree of freedom in the direct design procedure, through which the features of the reference model can be shaped based on real specifications (often given in less rigorous terms). Inspired by the framework introduced in (Formentin et al., 2016), we cast a constrained optimization problem, that allows one to easily incorporate *soft* specifications on the desired behavior (*e.g.,* settling time to step-like references, minimum bandwidth). Through successive relaxations and constraints lifting, we obtain a problem that resembles the one solved in (Breschi and Formentin, 2020) and highlight a direct relationship between the regularization strength exploited within the controller design phase and the parameters characterizing the obtained cost, thus allowing for their intuitive interpretation. The relaxed problem is then optimized with respect to the parameters of the reference model and the ones of the LPV controller via a nested procedure, combining Bayesian Optimization (BO) (Brochu et al., 2010) and the non-parametric approach introduced in (Formentin et al., 2016).

The paper is organized as follows. The data-driven control problem is introduced in Section 2, while its reformulation with softened constraints is discussed in Section 3. The proposed strategy is described in Section 4, and its effectiveness is assessed on a benchmark case study in Section 5. The paper is ended by some concluding remarks.

2. SETTING AND GOAL

Let $\mathcal{G}_p$ be a single-input single-output (SISO), minimum-phase, stable [1] LPV system, whose behavior is assumed to

⋆ This work was partially supported by the Italian Ministry of University and Research under the PRIN'17 project "Data-driven learning of constrained control systems", contract no. 2017J89ARP.

[1] Stability is intended as in (Formentin et al., 2016).

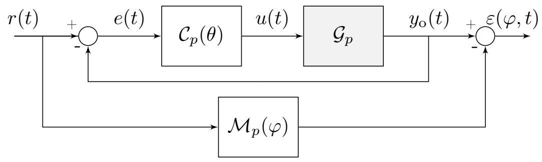

Fig. 1. Considered matching scheme.

be governed by the *unknown* difference equation

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

where $u(t)$ and $y_{\mathrm{o}}(t)$ are the input fed to the system at time $t \in \mathbb{N}$ and the corresponding noiseless output. Let $p(t) \in \mathbb{P} \subseteq \mathbb{R}^{n_p}$ be a set of scheduling variables and assume that the unknown coefficients $a_i(p,t)$ and $b_j(p,t)$ can be nonlinear dynamics mappings of the scheduling sequence, thus possibly depending on both current and past values of the scheduling signals.

In this work, we aim at designing a controller for the unknown plant $\mathcal{G}_p$, so as to satisfy *a set of user-defined specifications* on the closed-loop behavior, *e.g.,* bounds on the settling time to step-like references, while trading-off between the attained performance and the required control effort. Due to the LPV nature of the considered system, we focus on designing LPV controllers of the form

$$\mathcal{C}_p(\theta): \; u(t) = A_c(p,t,\theta,q^{-1})u(t) + B_c(p,t,\theta,q^{-1})e(t), \quad (2a)$$

where $e(t) = r(t) - y(t)$ denotes the tracking error at time t with respect to a customizable reference $r(t) \in \mathbb{R}$, and

$$A_c(p,t,\theta,q^{-1}) = \sum_{i=1}^{n_{a_c}} a_i^c(p,t,\theta) q^{-i}, \quad (2b)$$

$$B_c(p,t,\theta,q^{-1}) = \sum_{j=0}^{n_{b_c}} b_j^c(p,t,\theta) q^{-j}, \quad (2c)$$

are *fixed-order* polynomials of the shift operator q^{-1}, whose coefficients can be any nonlinear (possibly dynamic) bounded function of the set of the scheduling variables and the parameters $\theta \in \mathbb{R}^{n_\theta}$.

Not to impose the desired specifications on the unknown closed-loop, we introduce a class $\mathcal{M}_p(\varphi)$ of stable parametric LPV reference models, *i.e.,*

$$\mathcal{M}_p(\varphi): y_d(\varphi,t) = M(p,t,\varphi,q^{-1}) r(t), \quad (3)$$

where $\varphi \in \mathbb{R}^{n_\varphi}$, $y_d(\varphi,t) \in \mathbb{R}$ represents the desired response to a reference $r(t) \in \mathbb{R}$ and $M(p,t,\varphi,q^{-1})$ compactly denotes the LPV mapping between these two signals. We then try to satisfy the set of specifications by selecting a reference model that meets these requirements, while designing a controller in (2) to tightly match the behavior it dictates, according to the scheme in Fig.1. This entails the minimization of the mismatching error $\varepsilon(\varphi,t) = y_{\mathrm{o}}(t) - y_d(\varphi,t)$, $\forall t$.

Let us further assume that experiments can be performed on $\mathcal{G}_p$, so as to gather data-based insights on its behavior. Specifically, assume we can collect a dataset $\mathcal{D}_T = \{\mathcal{U}_T, \mathcal{P}_T, \mathcal{Y}_T\}$ comprising a set of inputs [2] $\mathcal{U}_T = \{u(t)\}_{t=1}^T$, a scheduling sequence $\mathcal{P}_T = \{p(t)\}_{t=1}^T$, and the corresponding noisy outputs $\mathcal{Y}_T = \{y(t)\}_{t=1}^T$, with

$$y(t) = y_{\mathrm{o}}(t) + v(t), \quad \forall t \in \{1,\ldots,T\}, \quad (4)$$

where $\{v(t)\}_{t=1}^T$ is a zero-mean, stationary (not necessarily white) noise sequence.

[2] The input sequence should be persistently exciting, see (Formentin et al., 2016).

Without involving a preliminary identification step, our goal translates into (*i*) automatically select a reference model in the class (3) to attain the user-defined specification, while (*ii*) learning a controller in (2) to achieve the desired closed-loop behavior. Given $\mathcal{D}_T$ and imposing that $u_c(\theta,0) = 0$ for all θ, we thus aim at solving the following design problem:

$$\min_{\theta,\varphi,\varepsilon} \sum_{t=1}^{T} \varepsilon(\varphi,t)^2 \quad (5a)$$

$$\text{s.t. } \varepsilon(t) = y(t) - y_d(\varphi,t), \quad \forall y(t) \in \mathcal{Y}_T, \quad (5b)$$

$$u(t) = u_c(\theta,t), \quad \forall u(t) \in \mathcal{U}_T, \quad (5c)$$

$$u_c(\theta,t) = A_c(p,t,\theta,q^{-1})u(t) + B_c(p,t,\theta,q^{-1})e(t), \quad \forall u(t) \in \mathcal{U}_T, p(t) \in \mathcal{P}_T, \quad (5d)$$

$$e(t) = r(t) - y(t), \quad \forall y(t) \in \mathcal{Y}_T, \quad (5e)$$

$$|y_d(\varphi,t) - r(t)| \leq \delta_1, \quad \forall t \in \mathcal{I}_T, \quad (5f)$$

$$|u_c(\theta,t) - u_c(\theta,t-1)| \leq \delta_2, \quad \forall t \in \mathcal{I}_T, \quad (5g)$$

$$\varphi \in [\underline{\varphi}, \overline{\varphi}], \quad (5h)$$

where we exploit the available data rather than introducing dependences on the (unknown) dynamics of the plant, $\underline{\varphi}, \overline{\varphi}$ are bounds on the parameters φ dictated by the user-defined specifications, while δ_1, δ_2 are positive constants. The considered cost function (5a) steers the mismatch between the available (noisy) outputs and the desired ones to zero (see (5b)). This goal has to be achieved by containing the desired tracking error according to (5f) and the required control effort (see (5g)). Note that the constraints in (5c)-(5e) entail that the one-step ahead input predicted with a controller in (2) has to match the measured input, for the definition of the mismatch error in (5b) to be consistent.

3. CONSTRAINTS SOFTENING FOR DIRECT LPV CONTROL DESIGN

The design problem (5) intrinsically depends on a reference $\{r(t)\}_{t=1}^T$ to be selected before-hand. This formulation requires one to leverage information other than the available data and the user-defined specifications. Moreover, by focusing on a single reference, the design procedure might result in a reference model, hence a controller, with poor generalization capabilities. To overcome this problem, we annihilate the dependence from this user-defined signal by replacing it with a *fictitious reference* $r_f(\varphi,t)$, obtained by exploiting the available data.

Let us assume that the mapping $M(p,t,\varphi,q^{-1})$ is invertible [3], namely there exists $M^\dagger(p,t,\varphi,q^{-1})$ such that

$$r(t) = M^\dagger(p,t,\varphi,q^{-1}) y_d(\varphi,t)$$

and $M^\dagger(p,t,\varphi,q^{-1}) M(p,t,\varphi,q^{-1}) = 1$. By manipulating the constraint in (5b), the fictitious reference can be defined as:

$$r_f(\varphi,t) = \tilde{y}(\varphi,t) - M^\dagger(p,t,\varphi,q^{-1})\varepsilon(\varphi,t), \quad (6)$$

[3] The reader is referred to (Formentin et al., 2016) for further details.

with $\tilde{y}(\varphi,t) = M^{\dagger}(p,t,\varphi,q^{-1})y(t)$. Problem (5) can thus be equivalently recast as follows:

$$\min_{\theta,\varphi,\varepsilon} \sum_{t=1}^{T} \varepsilon(\varphi,t)^2 \tag{7a}$$

$$\text{s.t. } u(t) = u_c(\theta,t), \quad \forall u(t) \in \mathcal{U}_T, \tag{7b}$$

$$u_c(\theta,t) = A_c(p,t,\theta,q^{-1})u(t) + B_c(p,t,\theta,q^{-1})e_f(\varphi,t), \quad \forall u(t) \in \mathcal{U}_T, p(t) \in \mathcal{P}_T, \tag{7c}$$

$$e_f(\varphi,t) = r_f(\varphi,t) - y(t), \quad \forall y(t) \in \mathcal{Y}_T, \tag{7d}$$

$$|y(t) - \varepsilon(\varphi,t) - r_f(\varphi,t)| \leq \delta_1, \quad \forall y(t) \in \mathcal{Y}_T, \tag{7e}$$

$$|u_c(\theta,t) - u_c(\theta,t-1)| \leq \delta_2, \quad \forall t \in \mathcal{I}_T, \tag{7f}$$

$$\varphi \in [\underline{\varphi}, \overline{\varphi}], \tag{7g}$$

where the first constraint of (5) is removed as it has been exploited to define the fictitious reference and the desired output in (7e), and the dependence of $u_c(\theta,t)$ on φ is omitted to simplify the notation. We remark that the assumptions on $M(p,t,\varphi,q^{-1})$ might entail restrictions on feasible parameterizations of the reference model.

Since the fictitious reference $r_f(\varphi,t)$ depends on the term $M^{\dagger}(p,t,\varphi,q^{-1})\varepsilon(\varphi,t)$ (see (6)), problem (7) features a set of nonlinear constraints. To remove them, we initially manipulate (7b)-(7d) and relax the constraints in (7e)-(7f) as detailed next. Let

$$\tilde{u}_c(\theta,t) = A_c(p,t,\theta,q^{-1})u(t) + B_c(p,t,\theta,q^{-1})\tilde{e}(\varphi,t), \tag{8a}$$

with $\tilde{e}(\varphi,t) = \tilde{y}(\varphi,t) - y(t)$ and $\tilde{y}(\varphi,t)$ defined based on (6). It can be easily shown that the following holds:

$$\tilde{\varepsilon}(\theta,\varphi,t) = \tilde{u}_c(\theta,t) - u(t), \tag{8b}$$

where $\tilde{\varepsilon}(\theta,\varphi,t) = B_c(p,t,\theta,q^{-1})M^{\dagger}(p,t,\varphi,q^{-1})\varepsilon(\varphi,t)$. By relying on this definition, along with the ones of $\tilde{u}_c(\theta,t)$ in (8a) and the fictitious reference, it can be shown that (7e) and (7f) can be respectively recast as:

$$|y(t) - \tilde{y}(\varphi,t) + (M^{\dagger}(p,t,\varphi,q^{-1}) - I)\varepsilon(\varphi,t)| \leq \delta_1, \tag{9a}$$

$$|\Delta\tilde{u}_c(\theta,t) + \Delta\tilde{\varepsilon}(\theta,\varphi,t)| \leq \delta_2, \tag{9b}$$

where

$$\Delta\tilde{u}_c(\theta,t) = \tilde{u}_c(\theta,t) - \tilde{u}_c(\theta,t-1), \tag{9c}$$

$$\Delta\tilde{\varepsilon}(\theta,\varphi,t) = \tilde{\varepsilon}(\theta,\varphi,t) - \tilde{\varepsilon}(\theta,\varphi,t-1). \tag{9d}$$

Based on the triangular inequality, we can relax these constraints so as to split the quantities depending on $\varepsilon(\varphi,t)$ from the other terms, as follows:

$$|y(t) - \tilde{y}(\varphi,t)| + |(M^{\dagger}(p,t,\varphi,q^{-1}) - I)\varepsilon(\varphi,t)| \leq \delta_1, \tag{10a}$$

$$|\Delta\tilde{u}_c(\theta,t)| + |\Delta\tilde{\varepsilon}(\theta,\varphi,t)| \leq \delta_2. \tag{10b}$$

This allows us to define the following relaxation of (7):

$$\min_{\theta,\varphi,\varepsilon} \sum_{t=1}^{T} \varepsilon(\varphi,t)^2 \tag{11a}$$

$$\text{s.t. } \tilde{\varepsilon}(\theta,\varphi,t) = \tilde{u}_c(\theta,t) - u(t), \quad \forall u(t) \in \mathcal{U}_T, \tag{11b}$$

$$\tilde{u}_c(\theta,t) = A_c(p,t,\theta,q^{-1})u(t) + B_c(p,t,\theta,q^{-1})\tilde{e}(\varphi,t), \quad \forall u(t) \in \mathcal{U}_T, p(t) \in \mathcal{P}_T, \tag{11c}$$

$$\tilde{e}(\varphi,t) = \tilde{y}(\varphi,t) - y(t), \quad \forall y(t) \in \mathcal{Y}_T, \tag{11d}$$

$$|y(t) - \tilde{y}(\varphi,t)| + |(M^{\dagger}(p,t,\varphi,q^{-1}) - I)\varepsilon(t)| \leq \delta_1, \quad \forall y(t) \in \mathcal{Y}_T, \tag{11e}$$

$$|\Delta\tilde{u}_c(\theta,t)| + |\Delta\tilde{\varepsilon}(\theta,t)| \leq \delta_2, \quad \forall t \in \mathcal{I}_T, \tag{11f}$$

$$\varphi \in [\underline{\varphi}, \overline{\varphi}], \tag{11g}$$

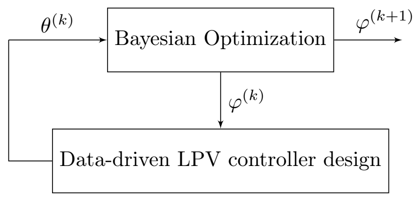

Fig. 2. Nested optimization: scheme of the kth iteration.

featuring a set of *nonlinear constraints*, where the products between $M^{\dagger}(p,t,\varphi,q^{-1})\varepsilon(\varphi,t)$ have been isolated.

By relying on this relaxed problem, we propose to minimize $\{\tilde{\varepsilon}(\theta,\varphi,t)\}_{t=1}^{T}$ instead of $\{\varepsilon(t)\}_{t=1}^{T}$, so that the cost depends on the difference between the measured and reconstructed inputs. This allows us to remove the constraint in (11b), but it demands for suitable techniques to counteract the effect of noise, so as to retrieve the solution obtained by minimizing $\varepsilon(\varphi,t)$ when the number of data goes to infinity. By minimizing $\{\tilde{\varepsilon}(\theta,\varphi,t)\}_{t=1}^{T}$, both the desired tracking error (see (11e)) and the control effort can be contained by minimizing variations of the input sequence $\{\tilde{u}_c(\theta,\varphi,t)\}_{t=1}^{T}$ and differences between the measured output $y(t)$ and $\tilde{y}(\varphi,t)$. Based on this intuition, we then soften and lift the constraints in (11e)-(11f), thus leading to a new data-driven design problem:

$$\min_{\theta,\varphi} J(\theta,\varphi) \tag{12a}$$

$$\text{s.t. } \varphi \in [\underline{\varphi}, \overline{\varphi}], \tag{12b}$$

$$\tilde{u}_c(\theta,t) = A_c(p,t,\theta,q^{-1})u(t) + B_c(p,t,\theta,q^{-1})\tilde{e}(\varphi,t), \quad \forall u(t) \in \mathcal{U}_T, p(t) \in \mathcal{P}_T, \tag{12c}$$

$$\tilde{e}(\varphi,t) = \tilde{y}(\varphi,t) - y(t), \quad \forall y(t) \in \mathcal{Y}_T, \tag{12d}$$

where

$$J(\theta,\varphi) = \sum_{t=1}^{T} \left[(y(t) - \tilde{y}(\varphi,t))^2 + W_u(\tilde{u}_c(\theta,t) - u(t))^2 + W_{\Delta u}\Delta\tilde{u}_c^2(\theta,t) \right], \tag{12e}$$

with W_u and $W_{\Delta u}$ being non-negative hyperparameters, that weight the fit of the inputs and the control effort, respectively. The penalization of the surrogate tracking error in the first term corresponds to a regularized difference between the reference model and a unitary gain, thus penalizing poor desired tracking performance. Note that problem (12) still features a nonlinear constraint in θ and φ, due to the dependence of (12c) on $\tilde{e}(\varphi,t)$.

4. DIRECT LPV CONTROL WITH FLEXIBLE REFERENCE MODELS

To avoid the solution of a nonlinear optimization problem and to leverage the dependence of the controller on the reference model [4], we tackle the control design problem in (12) through the nested approach sketched in Fig. 2. At a higher level, Bayesian Optimization (BO) (Brochu et al., 2010) is used to select the reference model to be tested, explicitly accounting for the soft specs dictated by the user. This derivative-free procedure is carried out by considering the dependence of the objective in (12e) on φ only, relying on modelling this cost as a Gaussian Process (GP) (Rasmussen and Williams, 2005) with mean $\mu(\varphi)$

[4] $\theta = \theta(\varphi)$ according to the constraint in (12c).

Algorithm 1 Flexible direct LPV control

Input: Dataset $\mathcal{D}_T$; bounds $\underline{\varphi}, \overline{\varphi}$; initial set $\mathcal{B}_\varphi^{(N)}$; k_{max}.

1. **for** $k = N, \dots, k_{max}$ **do**
   1.1 **update** the GP model of $J(\varphi)$ in (12e) via $\mathcal{B}_\varphi^{(k)}$;
   1.2 **find** the next candidate reference model as
   $$\varphi^{(k+1)} \leftarrow \arg \max_{\underline{\varphi} \leq \varphi \leq \overline{\varphi}} \alpha(\varphi | \mathcal{B}_\varphi^{(k)});$$
   1.3 **learn** the new LPV controller as
   $$\theta^{(k+1)} \leftarrow \arg \min_{\theta} J(\theta, \varphi^{(k+1)});$$
   1.4 **compute** $J(\theta^{(k+1)}, \varphi^{(k+1)})$ as in (12e);
   1.5 $\mathcal{B}_\varphi^{(k+1)} \leftarrow \mathcal{B}_\varphi^{(k)} \cup (\varphi^{(k+1)}, J(\theta^{(k+1)}, \varphi^{(k+1)}))$;
2. $(\theta^\star, \varphi^\star) \leftarrow \arg\min_{k=1,\dots,k_{max}} J(\theta^{(k)}, \varphi^{(k)})$.

Output: Reference model $\mathcal{M}_p(\varphi^\star)$ and controller $\mathcal{C}_p(\theta^\star)$.

and covariances $\sigma^2(\varphi, \varphi')$. At a lower lever, we retrieve the data-driven LPV controller for the fixed reference model. This nested routine allows us to automatically find a reference model/controller pair without requiring closed-loop tests, benefiting from state-of-the-art approaches and avoiding naïve (and likely inefficient) grid searches over the set of admissible reference models.

The main steps of the nested routine are summarized in Algorithm 1, where the cost in (12e) is sometimes indicated as $J(\varphi)$ with a slight abuse of notation. Let $\{\varphi^{(k)}\}_{i=1}^N$ be a set of (randomly) chosen feasible parameters of (3), and denote with $\{\theta^{(k)}\}_{i=1}^N$ the ones of the corresponding LPV controllers in (2). Starting from the initial set

$$\mathcal{B}_\varphi^{(N)} = \{\varphi^{(k)}, J(\theta^{(k)}, \varphi^{(k)})\}_{k=1}^N, \tag{13}$$

Algorithm 1 iteratively unfolds as follows. At step 1.1, the set of tested references and corresponding cost is used to update the GP model of $J(\varphi)$. The next reference model to be tested is then chosen by maximizing a user-defined *acquisition function* $\alpha(\varphi|\mathcal{B}_\varphi^{(k)})$, that balances exploration and exploitation of the search space dictated by $\underline{\varphi}$ and $\overline{\varphi}$, as in step 1.2. This reference model is used to train an LPV controller (see step 1.3). At steps 1.4-1.5, we compute the cost in (12e) associated to the reference model/controller pair and update the set $\mathcal{B}_\varphi^{(k)}$. This procedure is repeated until a prefixed number of iterations k_{max} is performed.

4.1 Towards direct nonparametric LPV control design

For each reference model $\varphi^{(k)}$ a new LPV controller has to be designed from data. In this work we focus on designing nonparametric controllers, so as to relieve the user from the need to select the structure of the controller beforehand. In this scenario, the coefficients in (2b)-(2c) are given by:

$$a_i^c(p, t, \theta) = \theta_i' \psi_i(p, t), \quad i = 1, \dots, n_{a_c}, \tag{14a}$$

$$b_j^c(p, t, \theta) = \theta_{\iota(j)}' \psi_{\iota(j)}(p, t), \quad j = 0, \dots, n_{b_c}, \tag{14b}$$

where $\iota(j) = n_{a_c} + j + 1$ for $j = 0, \dots, n_{b_c}$, $\psi_h(p, t) : \mathbb{P} \to \mathbb{R}^{n_H}$ is an *unknown* nonlinear function from the scheduling space to a feature space $\mathcal{R}^{n_H}$ of *unknown* dimension $n_H \in \mathbb{N}$, while $\theta_h \in \mathbb{R}^{n_H}$ is the *unknown* set of parameters to be learned, for $h = 1, \dots, n_f$ and $n_f = n_{a_c} + n_{b_c} + 1$. Therefore, by introducing the regressor $x(t) \in \mathbb{R}^{n_f}$:

$$x(t) = [-u(t-1) \cdots -u(t-n_{a_c}) \ \tilde{e}(t) \cdots \tilde{e}(t-n_{b_c})]', \tag{14c}$$

the constraint in (12c) can be recast as:

$$\tilde{u}_c(\theta, t) = \sum_{h=1}^{n_f} \theta_h' \psi_h(p, t) x_h(t), \tag{14d}$$

where $x_h(t)$ is the h-th component of $x(t)$ in (14c), and the dependence on $\varphi^{(k)}$ is not explicitly shown.

By considering this class of controllers, straightforward manipulations of the cost in (12e) allows us to cast the data-driven control design problem as follows:

$$\min_{\theta, \varepsilon_u} \sum_{t=1}^{T} \left[\frac{W_u}{W_{\Delta u} T^2} \varepsilon_u(t)^2 + \frac{1}{T^2} \Delta \tilde{u}_c(\theta, t)^2 \right] \tag{15a}$$

$$\text{s.t. } \varepsilon_u(t) = u(t) - \tilde{u}_c(\theta, t), \quad \forall u(t) \in \mathcal{U}_T, \tag{15b}$$

$$\tilde{u}_c(\theta, t) = \sum_{h=1}^{n_f} \theta_h' \psi_h(p, t) x_h(t), \tag{15c}$$

where (15c) holds $\forall \{y(t), u(t), p(t)\} \in \mathcal{D}_T$ and the second term in (15a) is a regularizer, that steers towards controllers resulting in smoother control actions. This formulation resembles an LS-SVM one (Suykens et al., 2002). In the following, we show (15) can indeed be recast as an LS-SVM problem by upper-bounding the regularizer.

Due to the structure of $\mathcal{C}_p(\theta)$ (see (14)), it holds that

$$\Delta \tilde{u}_c(\theta, t) = \sum_{h=1}^{n_f} \theta_h' \left[\psi_h(p, t) x_h(t) - \psi_h(p, t-1) x_h(t-1) \right].$$

By introducing the vector $\Theta_\psi(t) \in \mathbb{R}^{1 \times n_f}$ defined as

$$\Theta_\psi(t) = [\theta_1' \psi_1(p, t) \ \cdots \ \theta_{n_f}' \psi_{n_f}(p, t)], \quad t = 1, \dots, T, \tag{16a}$$

and the positive semi-definite matrix

$$\mathcal{X}(t, t-1) = \begin{bmatrix} x(t)x(t)' & -x(t)x(t-1)' \\ -x(t-1)x(t)' & x(t-1)x(t-1)' \end{bmatrix}, \tag{16b}$$

it can be easily shown that the following holds:

$$\Delta \tilde{u}_c^2(\theta, t) \leq \bar{\lambda}(t, t-1) [\Theta_\psi(t) \ \Theta_\psi(t-1)] \begin{bmatrix} \Theta_\psi(t)' \\ \Theta_\psi(t-1)' \end{bmatrix}, \tag{16c}$$

where $\bar{\lambda}(t, t-1) > 0$ is the greatest eigenvalue of $\mathcal{X}(t, t-1)$. Thanks to the characteristics of (16c), we can then remove the dependence of the upper-bound on the unknown $\{\psi_h(p, t)\}_{h=1}^{n_f}$ by exploiting the *kernel trick* (Vapnik, 1998), *i.e.*, replacing the inner products $\langle \psi_h(p, t), \psi_h(p, \tau) \rangle$ with user-defined *kernel functions* $\kappa_h(p, t, \tau)$, for $t, \tau = 1, \dots, T$ and $h = 1, \dots, n_f$. This allows us to define

$$c = \max_{h=1,\dots,n_f} \left\{ \frac{1}{T^2} \sum_{t=1}^{T} \bar{\lambda}(t, t-1) \tilde{\kappa}_h(p, t, t-1) \right\}, \tag{16d}$$

with $\tilde{\kappa}_h(p, t, t-1) = \kappa_h(p, t, t) + \kappa_h(p, t-1, t-1)$, that provides an upper-bound to the right-hand-side of (16c).

Accordingly, we can cast the following relaxed design problem in LS-SVM form as:

$$\begin{aligned} &\min_{\theta, \varepsilon_u} \sum_{h=1}^{n_f} \left[\theta_h' \theta_h + \frac{\gamma}{T^2} \sum_{t=1}^{T} \varepsilon_u(t)^2 \right] \\ &\text{s.t. } \varepsilon_u(t) = u(t) - \sum_{h=1}^{n_f} \theta_h' \psi_h(p, t) x_h(t), \end{aligned} \tag{17}$$

where the constraint has to hold for all $\{y(t), u(t), p(t)\} \in \mathcal{D}_T$ and $\gamma = \frac{W_u}{c W_{\Delta u}}$, thus showing the existence of a relationship between the weights in (12) and the regularization strength. We stress that the upper-bounds used to retrieve problem (17) act on the regularization term only. It is thus expected that a proper choice of the weight W_u and $W_{\Delta u}$ can balance the effect of the over-approximation.

4.2 Learning the LPV controller

To cope with the effect of measurement noise we exploit the *instrumental variable* (IV) scheme presented in (Laurain et al., 2012), rather than solving (17). To this end, let $\tilde{x}_h(t)$ be a new realization of the regressor defined in (14c) and let $\zeta_h = \psi(p,t)\tilde{x}_h(t)$ for $h = 1,\dots,n_f$ and $t = 1,\dots,T$, be the set of associated instrumental variables. Problem (17) is thus ultimately recast as

$$\min_{\theta,\varepsilon_u} \sum_{h=1}^{n_f} \left[\theta_h'\theta_h + \frac{\gamma}{T^2}\left\|\sum_{t=1}^{T}\zeta_h(t)\varepsilon_u(t)\right\|_2^2\right] \quad (18a)$$

$$\text{s.t. } \varepsilon_u(t) = u(t) - \sum_{h=1}^{n_f}\theta_h'\psi_h(p,t)x_h(t), \quad (18b)$$

with the constraint holding for all $\{y(t), u(t), p(t)\} \in \mathcal{D}_T$.

Thanks to the features of problem (18), the global optimum can be found via the *Karush-Kuhn-Tucker* (KKT) conditions on the associated Lagrangian. This step entails the introduction of T Lagrange multipliers $\{\delta(t)\}_{t=1}^{T}$, that can be computed by using the kernel trick as

$$\delta = R_D(\Omega)^{-1}\frac{1}{T^2}\sum_{h=1}^{n_f}\tilde{X}_h\Omega_h\tilde{X}_hU, \quad (19a)$$

where $\delta, U \in \mathbb{R}^T$ respectively stack the Lagrange multipliers and the inputs, $\Omega_h \in \mathbb{R}^{T\times T}$ is a matrix whose element in position (t,τ) is equal to $\kappa_h(p,t,\tau)$, for $h = 1,\dots,n_f$, and $R_D(\Omega) \in \mathbb{R}^{T\times T}$ in (19a) is given by

$$R_D(\Omega) = \gamma^{-1}I + \frac{1}{T^2}\sum_{h=1}^{n_f}\tilde{X}_h\Omega_h\tilde{X}_h\sum_{j=1}^{n_f}X_j\Omega_iX_j, \quad (19b)$$

with $X_h, \tilde{X}_h \in \mathbb{R}^{T\times T}$ being diagonal matrices with t-th entries respectively corresponding to $x_h(t)$ and $\tilde{x}_h(t)$. From the multipliers, the coefficients in (14) can be retrieved as

$$a_i^c(p,t,\theta) = \sum_{\tau=1}^{T}\kappa_i(p,\tau,t)x_i(\tau)\delta(\tau), \quad i = 1,\dots,n_{a_c}, \quad (20)$$

$$b_j^c(p,t,\theta) = \sum_{\tau=1}^{T}\kappa_{\iota(j)}(p,\tau,t)x_{\iota(j)}(\tau)\delta(\tau), \quad (21)$$

with $\iota(j) = n_{a_c} + j + 1$, for $j = 0,\dots,n_{b_c}$.

Remark 1. The user-defined *kernels* $\{\kappa_h(p,t,\tau)\}_{h=1}^{n_f}$ dictate the shape of the learned LPV controller both directly and by influencing the regularization strength (see (16d)). As such, they have to be chosen with care to attain satisfactory closed-loop performance. ■

5. A BENCHMARK CASE STUDY

The effectiveness of the proposed design strategy is assessed by considering a voltage-driven DC motor that features an inhomogeneous mass distribution, as in (Formentin et al., 2016). This servo-positioning system is described by the set of differential equations:

$$\begin{bmatrix}\dot{\theta}(t)\\ \dot{\omega}(t)\\ \dot{I}(t)\end{bmatrix} = \begin{bmatrix}0 & 1+\frac{\sin(\theta(\tau))}{\theta(\tau)} & 0\\ \frac{Mgl}{J}\frac{\sin(\theta(\tau))}{\theta(\tau)} & -\frac{b}{J} & \frac{K}{J}\\ 0 & -\frac{K}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\theta(t)\\ \omega(t)\\ I(t)\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{1}{L}\end{bmatrix}V(t),$$

where $\theta(t)$ [rad] is the *measured* angular position of the motor, $\omega(t)$ [rad/s] is its angular velocity, $I(t)$ [mA] is the current flowing in the motor circuit, $V(t)$ [V] is the controlled voltage, and the remaining parameters are the ones introduced in (Kulcsár et al., 2009). Note that, $\mathcal{G}_p$ is a *quasi-LPV system* with $p(t) = y(t)$.

For the results obtained with the proposed *flexible* framework to be comparable with the ones achieved in (Formentin et al., 2016), we consider the same controller structure and experimental settings used therein. Therefore, we set $n_{a_c} = n_{b_c} = 4$ in (2), we replace the dependence on the tracking error with the one on its integral $e_{int}(t)$, with $e_{int}(t) = e_{int}(t-1) + (r(t) - y(t))$, and we impose

$$a_i^c(p,t,\theta) = a_i^c(\Pi(t),\theta), \ \forall i = 1,\dots,n_{a_c}, \quad (22a)$$

$$b_j^k(p,t,\theta) = b_j^k(\Pi(t),\theta), \ \forall j = 0,1,\dots,n_{b_c}, \quad (22b)$$

with $\Pi(t) = [p(t-1)\ p(t-2)\ p(t-3)\ p(t-4)]'$. The dataset $\mathcal{D}_T$ comprises $T = 1500$ samples, gathered by feeding the DC motor with a filtered Gaussian distributed white noise sequence with mean 16 V. The instrument needed in (18) is retrieved via an additional experiment with this input sequence. The output is corrupted by a zero-mean white noise with Gaussian distribution, yielding a *Signal-to-Noise Ratio* (SNR) of 43 dB. The data are acquired via a *zero-order-hold* scheme with sampling time 0.01 s, with a anti-aliasing filter acting on the output. As in (Formentin et al., 2016), we focus on first order LTI reference models

$$\mathcal{M}(\varphi):\ y_d(\varphi,t) = \frac{q^{-1}(1-e^{\varphi})}{1-q^{-1}e^{\varphi}}r(t), \quad (23)$$

with $\varphi \in [-0.5, -0.005]$, to enforce a settling time to step-like references in $[0.1, 10]$ s. Note that, when $\varphi \approx -0.01$, (23) is the model considered in (Formentin et al., 2016).

The nested approach is run for different combinations of the weights $(W_u, W_{\Delta u})$, by performing 50 BO iterations, with $\alpha(\varphi|\mathcal{B}_\varphi)$ at step 1.2 being the *Expected Improvement* (EI) and by using Matérn kernels (Neal, 2012) to update the model of (12e). *Radial Basis Functions* (RBF) kernels with fixed width of 2.4 are used to learn the controller as in (Formentin et al., 2016). The achieved closed-loop performance is assessed by testing the resulting controllers on a realistic simulator. Let $y_{cl}^\star(t)$ be the simulated closed-loop output and $y_d^\star(t)$ be the desired response of the automatically chosen reference model at time t. We consider the following figures:

$$\text{RMSE}_M(\theta^\star,\varphi^\star) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y_{cl}^\star(t) - y_d^\star(t)\right)^2}, \quad (24a)$$

$$\text{RMSE}_y(\theta^\star,\varphi^\star) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_{cl}^\star(t) - r(t))^2}, \quad (24b)$$

that provide quantitative insights on the quality of match with the desired behavior and the tracking performance, respectively. The results in Fig. 3 and Fig. 4 show that the region where better performance is achieved in terms of the two indicators in (24) corresponds to relatively

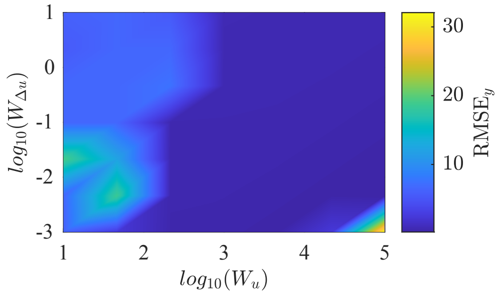

Fig. 3. $\mathrm{RMSE}_y(\theta^\star, \varphi^\star)$ *vs* $(W_u, W_{\Delta u})$. The dark blue region corresponds to satisfactory performance in terms of reference model and output tracking, since $\mathrm{RMSE}_M(\theta^\star, \varphi^\star)$ exhibits the same behavior.

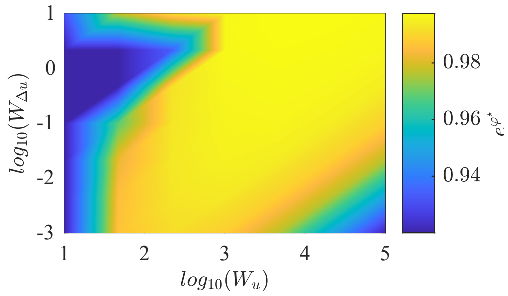

Fig. 4. Selected pole of the reference model in (23) *vs* $(W_u, W_{\Delta u})$. The yellow region corresponds to poles within the interval $[0.965,\ 0.9975]$.

Table 1. Flexible *vs* fixed reference model

	$\exp(\varphi^\star)$	γ	RMSE_y	RMSE_M
Flexible	0.977	$1.2328\ 10^6$	0.246	0.053
Fixed	0.9900	64163	0.353	0.049

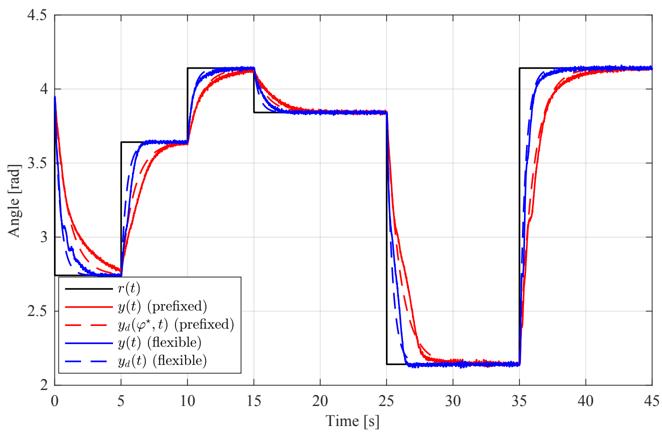

Fig. 5. Flexible (blue) *vs* fixed (red) (Formentin et al., 2016) reference model. Desired behavior (dashed lines) and attained closed-loop response (solid lines).

slow dynamics characterizing the chosen reference model, showing that a proper choice of W_u and $W_{\Delta u}$ allows one to account for the attainability of the desired behavior. We finally compare the result obtained for a fixed pair of weights $(W_u, W_{\Delta u})$ and the ones achieved in (Formentin et al., 2016), where the reference model is conservatively *fixed before-hand*. As expected (see Table 1 and Fig. 5) with a flexible reference model we visibly improve the tracking performance, without compromising model matching.

6. CONCLUSIONS

In this paper, we presented an approach for the direct design of LPV controllers with flexible reference models. By treating the latter as a constrained *hyper-parameter*, we require the user to select the structure of the reference model and provide some bounds on the desired closed-loop behavior, rather than fully specify it. Future research will be devoted to extend the approach to non-minimum phase and unstable plants.

REFERENCES

Berberich, J., Köhler, J., Müller, M., and Allgöwer, F. (2021). Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 66(4), 1702–1717.

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

Brochu, E., Cora, V.M., and de Freitas, N. (2010). A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. URL `http://arxiv.org/abs/1012.2599`.

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

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

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

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

Kulcsár, B., Dong, J., van Wingerden, J.W., and Verhaegen, M. (2009). LPV Subspace Identification of a DC motor with unbalanced disc. *IFAC Proceedings Volumes*, 42(10), 856–861.

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

Lecchini, A. and Gevers, M. (2002). On iterative feedback tuning for non-minimum phase plants. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 4, 4658 – 4663.

Neal, R.M. (2012). *Bayesian Learning for Neural Networks*. Springer New York.

Rasmussen, C.E. and Williams, C.K.I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Selvi, D., Piga, D., and Bemporad, A. (2018). Towards direct data-driven model-free design of optimal controllers. In *2018 European Control Conference (ECC)*, 2836–2841.

Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. (2002). *Least Squares Support Vector Machines*. WORLD SCIENTIFIC.

Vapnik, V.N. (1998). *Statistical Learning Theory*. Wiley-Interscience.