Proceedings of Machine Learning Research vol 120:1-9, 2020

Virtual Reference Feedback Tuning with data-driven reference model selection

Valentina Breschi VALENTINA.BRESCHI@POLIMI.IT
Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy.

Simone Formentin SIMONE.FORMENTIN@POLIMI.IT
Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy.

Abstract

In control applications where finding a model of the plant is the most costly and time consuming task, Virtual Reference Feedback Tuning (VRFT) represents a valid - purely data-driven - alternative for the design of model reference controllers. However, the selection of a proper reference model within a model-free setting is known to be a critical task, with this model typically playing the role of a hyper-parameter. In this work, we extend the VRFT methodology to compute both a proper reference model and the corresponding optimal controller parameters from data by means of Particle Swarm optimization. The effectiveness of the proposed approach is illustrated on a benchmark simulation example.

Keywords: VRFT, model reference control, data-driven control

1. Introduction

Virtual Reference Feedback Tuning (VRFT) is a model-reference control design technique, in which a parametric controller is tuned directly using a set of experimental data, see Campi et al. (2002); Formentin et al. (2019). As compared to more traditional model-based methodologies, pure data-driven approaches have many advantages, *e.g.,* specific system expertise is no longer required, the time needed for control design is largely reduced, as efforts are usually mainly dedicated to system modeling, to cite a few, and, under some conditions, the obtained performance might be significantly better than the one achieved with model-based alternatives, *e.g.,* see Formentin et al. (2014).

Nevertheless, like all model-reference approaches, VRFT requires the definition of a reference model to compute the controller parameters. Such a model should be achievable when closing the loop with a controller within the considered parametric class, since a bad choice of the desired closed-loop would lead to poor performance or instabilities, as illustrated in Nijmeijer and Savaresi (1998); van Heusden et al. (2011). It follows that, if the best achievable performance is not easily computable (which is likely to happen especially when a model of the system is not available), the reference model becomes a tuning knob or a *hyper-parameter*, which may need significant additional effort to be selected. For the above reasons, some techniques have already been proposed in the literature to properly select a suitable reference model from data. In Selvi et al. (2018), this model is selected as the one that leads to the controller that best tracks a pre-defined reference trajectory. Instead, in Kergus et al. (2019), a set of proper reference models is selected via interpolation constraints in the Lowner framework for data-driven control of Kergus et al. (2017, 2018).

Similarly to Selvi et al. (2018), we set an optimization problem for the joint design of the reference model and the controller. However, unlike Selvi et al. (2018), in our approach we do not stick to a specific reference signal, as we exploit the VRFT rationale also for reference model selection. Particle swarm optimization (see Poli et al. (2007)) will be used as a tool to find the optimal solution. The advantages over the existing techniques will be illustrated on a benchmark numerical example.

The paper is structured as follows. Section 2 introduces the considered problem, setting the ground for the proposed solution for combined reference model and controller design presented in Section 3. The performance attained with the proposed data-driven approach are illustrated in Section 4, by means of a numerical case study. The paper is ended by some concluding remarks.

2. Problem setting

Let P be an (unknown) *single-input single-output* (SISO) system. When the plant is fed with an input u, we assume that the (noisy) output y can be measured. This allows us to construct a dataset $\mathcal{D}_T = \{u(t), y(t)\}_{t=1}^{T}$ of input/output measurements that, under the assumption that can be used to directly estimate a controller C for the system, rather than first identifying a model P and then devising a model-based controller, under the assumption that the input sequence $\{u(t)\}_{t=1}^{T}$ is persistently exciting.

Throughout the rest of the paper, we focus on designing a linear discrete-time control law with *fixed structure* and embedded integral action, namely

$$u_c(\theta, t) = C(\theta, q)(r(t) - y(t)), \tag{1}$$

where q is the forward shift operator (*i.e.,* $qu(t) = u(t+1)$), $r(t) \in \mathbb{R}$ is a reference signal to be tracked and $\theta \in \mathbb{R}^{n_\theta}$ is a set of (unknown) parameters to be learned from data. The controller is designed so as to optimize the following performance criterion

$$\mathcal{J}(\theta) = \frac{1}{T}\sum_{t=1}^{T} W_e(r(t) - y(t))^2 + W_{\Delta u}\Delta^2 u_c(\theta) + W_u(u(t) - u_c(\theta, t))^2, \tag{2}$$

with Δu_c being the difference between two consecutive inputs and the positive weights $W_e, W_{\Delta u}$ and W_u tuned to trade off between three concurrent goals, namely the reduction of (i) the tracking error, (ii) the actuator effort and (iii) the error between the training input u and the reconstructed control action u_c. Note that this performance index resembles the one typically used for performance-driven control (see Bemporad et al. (2004)), with the last term introduced to account for the data-driven nature of our solution. Since we do not want to identify a model of the plant, we will study how to find a controller with the specified parameterization by exploiting the VRFT approach of Campi et al. (2002).

3. VRFT with data-driven reference model selection

The VRFT approach relies on the definition of a reference model $M(\varphi, q)$ describing the desired closed-loop behavior, whose parameters $\varphi \in \mathbb{R}^{n_\varphi}$ are usually fixed by the user beforehand. However, the selected reference model might heavily impact the actual performance measured by (2), especially if the user-defined desired behavior is too demanding. Instead of fixing the reference model a priori, in this paper we propose to look at it as a *hyper-parameter* and learn its parameters φ so to optimize the more general criterion in (2).

Let $y_d(\varphi, t) \in \mathbb{R}$ being the output of $M(\varphi, q)$ when fed with the reference $r(t)$, *i.e.,*

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

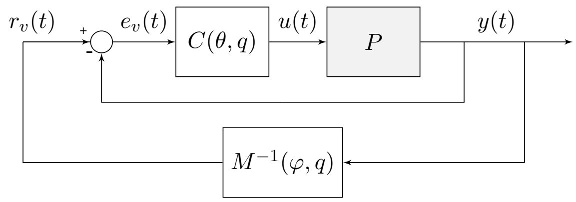

Figure 1: VRFT scheme, with the unknown plant highlighted in gray.

For a fixed model $M(\varphi, q)$, suppose we have designed a control law as in (1) with parameters $\theta(\varphi)$ through VRFT. Let $u_d(\theta(\varphi), t)$ be the input resulting from the designed controller when fed with the reference tracking error $r(t) - y_d(\varphi, t)$, namely

$$u_d(\theta(\varphi), t) = C(\theta(\varphi), q)(r(t) - y_d(\varphi, t)), \tag{4}$$

and define as $u_c(\theta(\varphi), t)$ the input resulting from the open-loop simulation of the controller based on the available data. Similarly to Selvi et al. (2018), the control design problem can be recast as follows

$$\min_{\theta,\varphi} \; \tilde{\mathcal{J}}(\theta, \varphi), \tag{5a}$$

where the objective is given by

$$\tilde{\mathcal{J}}(\theta, \varphi) = \frac{1}{T}\sum_{t=1}^{T} W_e(r(t) - y(t))^2 + W_{\Delta u}\Delta^2 u_d(\theta(\varphi)) + W_u(u(t) - u_c(\theta(\varphi), t))^2, \tag{5b}$$

with $\Delta u_d(\theta(\varphi))$ being the difference between two consecutive values of the *reference* input u_d. Note that the cost in (5b) depends on the unknowns of both the controller and the reference model.

By looking at (5b), it can be seen that the objective function still depends on the reference to be tracked r, thus implying that this signal has to be chosen beforehand, at the cost of possible over-fitting specific scenarios. This somehow diverges from the philosophy behind the VRFT method, where $M(\varphi, q)$ is exploited to construct a *virtual reference* r_v such that

$$y(t) = M(\varphi, q) r_v(t), \tag{6}$$

where $y(t)$ is the output measured at time t (see Figure 1). This fictitious set point is introduced not to fix the reference to be tracked beforehand, thus avoiding over-fitting, and it is indeed a function of the (unknown) parameters φ. By replacing the fixed reference r with the virtual one $r_v(\varphi, t)$, the cost function is thus modified as follows

$$\tilde{\mathcal{J}}_v(\theta, \varphi) = \frac{1}{T}\sum_{t=1}^{T} W_e(r_v(\varphi, t) - y(t))^2 + W_{\Delta u}\Delta u_{d,v}(\theta(\varphi))^2 + W_u(u(t) - u_{c,v}(\theta(\varphi), t))^2, \tag{7a}$$

where $u_{d,v}(\theta(\varphi), t)$ is the input given by the estimated controller $C(\theta(\varphi), q)$ when considering the virtual reference and closing the loop on the reference model, *i.e.,*

$$u_{d,v}(\theta(\varphi), t) = C(\theta(\varphi), q)(r_v(t) - y_{d,v}(\varphi, t)), \tag{7b}$$

with

$$y_{d,v}(\varphi, t) = M(\varphi, q) r_v(\varphi, t), \tag{7c}$$

and $u_{c,v}(\theta(\varphi), t)$ is the control input computed from the available data, namely

$$u_{c,v}(\theta(\varphi,)t) = C(\theta(\varphi), q)(r_v(\varphi, t) - y(t)). \tag{7d}$$

By the definition of the virtual reference $r_v(\varphi, t)$, note that $y_{d,v}(t)$ corresponds to the available output measurements and, as a consequence, $u_{d,v}(\theta(\varphi,)t) = u_{c,v}(\theta(\varphi,)t)$. in our setting we thus weight the variations of the controller input. Differently from Selvi et al. (2018), the objective in (7a) depends only on the available data and on the chosen parametrization of the controller and the reference model, since the virtual reference is constructed based on data.

Note that the derivation of the virtual reference from (6) requires the computation of the left inverse $M^{-1}(\varphi, q)$ of $M(\varphi, q)$ (such that $M^{-1}(\varphi, q)M(\varphi, q) = 1$), since it holds that

$$r_v(\varphi, t) = M^{-1}(\varphi, q)y(t). \tag{8}$$

This operation might be particularly involved for nonlinear reference models, *e.g.,* when considering *linear parameter varying* reference models Formentin et al. (2016). To overcome this limitation and to ease the extension of the approach to more complex settings, we propose to learn directly the left inverse $M^{-1}(\varphi, q)$. In this work we assume the reference model to be parametrized according to the following zero-pole-gain representation

$$M(\varphi, q) = K_M \frac{\Pi_{i=1}^{n_z^r}(q - z_i^r)\Pi_{j=1}^{n_z^c}\left(q - z_j^{c,re} \pm jz_j^{c,im}\right)}{\Pi_{i=1}^{n_p^r}(q - p_i^r)\Pi_{j=1}^{n_p^c}\left(q - p_j^{c,re} \pm jp_j^{c,im}\right)}, \tag{9}$$

so that φ stacks all the (unknown) poles and zeros of the reference model, and the corresponding left inverse is given by

$$M^{-1}(\varphi, q) = K_{M^{-1}} \frac{\Pi_{i=1}^{n_p^r}(q - p_i^r)\Pi_{j=1}^{n_p^c}\left(q - p_j^{c,re} \pm jp_j^{c,im}\right)}{\Pi_{i=1}^{n_z^r}(q - z_i^r)\Pi_{j=1}^{n_z^c}\left(q - z_j^{c,re} \pm jz_j^{c,im}\right)}, \tag{10}$$

with $K_{M^{-1}} = K_M^{-1}$. Let

$$n_z = n_z^r + n_z^c, \quad n_p = n_p^r + n_p^c,$$

be the *fixed* numbers of overall zeros and poles of the reference model. For $M(\varphi, q)$ to be physically meaningful, the number of poles and zeros is selected so that $n_p \geq n_z$. However, when the reference model is strictly proper, the left inverse defined as in (10) is no longer realizable. To overcome this limitation, we introduce the filter

$$L_r(q) = \frac{1}{q^{n_p - n_z}}, \tag{11}$$

which is used to properly modify the first term in the cost, that, in turns, depends on the left inverse $M^{-1}(\varphi, q)$ explicitly. The final design problem is thus given by

$$\min_{\theta, \varphi} \ \tilde{\mathcal{J}}_{v,L_r}(\theta, \varphi), \tag{12a}$$

with

$$\tilde{\mathcal{J}}_{v,L_r}(\theta, \varphi) = \frac{1}{T}\sum_{t=1}^{T} W_e(r_{v,L_r}(\varphi, t) - y_{L_r}(t))^2 + W_{\Delta u}\Delta^2 u_{c,v}(\theta(\varphi)) + W_u(u(t) - u_{c,v}(\theta(\varphi), t))^2 \tag{12b}$$

and

$$r_{v,L_r}(\varphi, t) = L_r(q) \cdot M^{-1}(\varphi, q)y(t), \tag{12c}$$

$$y_{L_r}(t) = L_r(q)y(t), \tag{12d}$$

being the filtered virtual reference and output, respectively.

Similarly to what is done in Selvi et al. (2018), the gain $K_{M^{-1}}$ is not optimized, but it is tuned once the parameters φ are chosen so that $M^{-1}(\varphi, 1) = 1$. To enforce stability on both the reference model and its left inverse, the cost in (12b) is further augmented with a set of barrier function penalizing instable reference configurations. As in Selvi et al. (2018), we introduce

$$h_j^p(\theta) = |p_j|^2 - 1 < 0, \;\; j = 1, \dots, n_p, \tag{13a}$$

$$h_i^z(\theta) = |z_i|^2 - 1 < 0, \;\; i = 1, \dots, n_z, \tag{13b}$$

with $|p_j|$ and $|z_i|$ indicating the modulus of the j-th pole and i-th zero of $M^{-1}(\varphi, q)$, respectively, and we consider the piecewise polynomial barrier functions $b : \mathbb{R} \to \mathbb{R}$,

$$b(h) = \begin{cases} 0, \text{ if } h < 0, \\ kh, \text{ if } 0 \leq h < 1, \\ kh^2, \text{ otherwise,} \end{cases} \tag{13c}$$

with $k \in \mathbb{R}_{>0}$ fixed by the user. The overall cost is thus given by

$$\tilde{\mathcal{J}}_{v,L_r}^b(\theta, \varphi) = \tilde{\mathcal{J}}_{v,L_r}(\theta, \varphi) + \sum_{i=1}^{n_p} b(h_i^p(\theta)) + \sum_{j=1}^{n_z} b(h_i^z(\theta)), \tag{13d}$$

with $\tilde{\mathcal{J}}_{v,L_r}(\theta, \varphi)$ defined as in (12b). We stress that, in case of complex numbers, the barrier function acts on both the real and the complex part of the singularity.

Algorithm 1 summarizes the whole procedure. Note that the controller is trained by using *particle swarm optimization* (PSO) as described in Poli et al. (2007), due to the nested dependence between the unknown parameters of the controller, the ones of the reference model and the virtual reference. The proposed approach involves the alternate minimization of the cost function in (13d) with respect to the parameters φ of M^{-1} (for a fixed controller $C(\theta, q)$), and the optimization of $\tilde{\mathcal{J}}_{v,L_r}^b(\theta, \varphi)$ with respect to θ for a fixed reference model. Although $\Delta u_{d,v}(\varphi(\theta))$ depends on the designed controller, the term weighting variations in the input over two consecutive steps is introduced to trade-off between the reference model performance and its actual attainability. Therefore, this term will be neglected when optimizing the cost $\tilde{\mathcal{J}}_v^b(\theta, \varphi)$ with respect to θ. The only term left in the cost is the one weighting the error in the reconstruction of the input, which correspond to the objective traditionally minimized in VRFT (see step 2.1.3).

Remark 1 *The weights W_e, $W_{\Delta u}$ and W_u have to be tuned by the user. Although this choice might be as challenging as a trial-and-error selection of M, by looking at* (10) *it becomes clear that the more complex is the reference model, the easier for the user is to select the (at most) two weights in the cost* (13d)*, as compared to the selection of all the parameters φ of M.*

4. Numerical example

The performance attained with the presented approach is assessed by designing a data-driven control law for the system shown in Figure 2 taken from Caré et al. (2019). The relationship between the force u applied to the system and the position y of m_2 is given by

$$P(s) = \frac{m_1 s^2 + (c_1 + c_2)s + (k_1 + k_2)}{(m_1 s^2 + (c_1 + c_2)s + k_1 + k_2)(m_2 s^2 + c_2 s + k_2) - (c_2 s + k_2)^2}, \tag{15}$$

Algorithm 1 [VRFT with data-driven reference model selection]

Input: Dataset $\mathcal{D}_T = \{u(t), y(t)\}_{t=1}^{T}$; number of particles N_p; barrier function b; weights W_e, $W_{\Delta u}$ and W_u; maximum number of iterations i_{max}.

1. **populate** particle swarms $\{\varphi^{(n)}\}_{n=1}^{N_p}$, so that the constraints in (13a) are satisfied.
2. **for** $i = 1, \dots, i_{max}$**do**
   - 2.1. **for** $n = 1, \dots, N_p$ **do**
     - 2.1.1. **fix** $M^{-1}(\varphi^{(n)}, q)$ based on (10);
     - 2.1.2. **select** $K_{M^{-1}}^{(n)}$ so that $M^{-1}(\varphi^{(n)}, 1) = 1$;
     - 2.1.3. **design** $C(\theta(\varphi^{(n)}))$ with VRFT Campi et al. (2002);
     - 2.1.4. **compute** the cost $\tilde{\mathcal{J}}_{v,L_r}^{b}(\theta^{(n)}, \varphi^{(n)})$ as in (13d), with $\theta^{(n)} = \theta(\varphi^{(n)})$ obtained at step 2.1.3;
   - 2.2. **end for**;
3. **choose** $(\theta^\star, \varphi^\star)$ such that

$$(\theta^\star, \varphi^\star) = \arg\min_{n=1,\dots,N_p} \tilde{\mathcal{J}}_{v,L_r}^{b}(\theta^{(n)}, \varphi^{(n)}); \tag{14}$$

4. **update** the particles as in (Poli et al., 2007, Algorithm 1);
5. **end for**.

Output: Left inverse parameters $\varphi^\star$, controller parameters $\theta^\star = \theta(\varphi^\star)$.

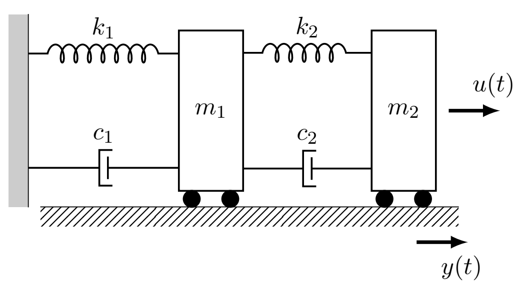

Figure 2: Scheme of the physical system to be controlled.

Table 1: Parameters of the system P.

m_1 [kg]	m_2 [kg]	c_1 [N/m^2]	c_2 [N/m^2]	k_1 [N/m]	k_2 [N/m]
1	0.5	0.2	0.5	1	0.5

where $s \in \mathbb{C}$ is the Laplace variable and the parameters are reported in Table 1. A set of input/output samples of length 5000 is generated by exciting the system with a zero-mean normally distributed input signal $u \sim \mathcal{N}(0, 1)$, with the corresponding measured output corrupted by an additive zero-mean Gaussian noise with variance 10^{-4}. Let $T_s = 0.1$ s be the sampling time of the system. As in

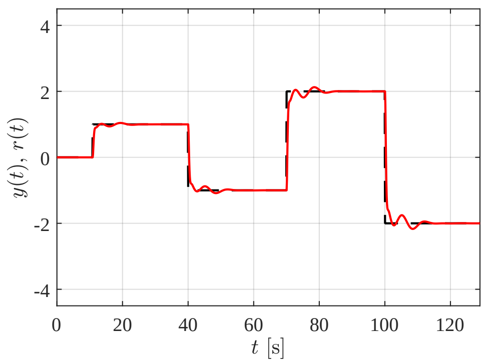

Figure 3: Reference signal (black) vs attained (noiseless) closed-loop output (dashed dotted red).

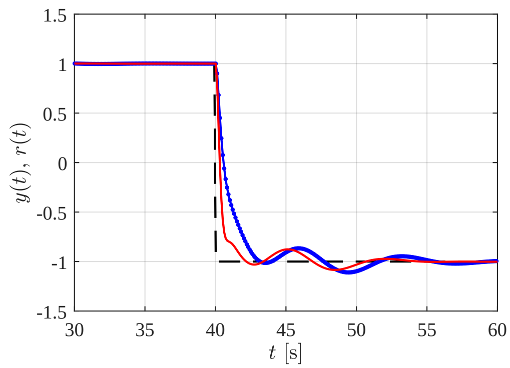

Figure 4: Reference signal (black) vs attained (noiseless) closed-loop outputs (detail). Data-driven model reference tuning+controller design (dashed dotted red), VRFT with fixed reference model (17) (dashed blue).

Caré et al. (2019), we aim at training a *Proportional-Integral-Derivative* (PID) controller, namely

$$C_{PID}(q^{-1}) = K_p + K_i \frac{T_s}{2} \frac{1+q^{-1}}{1-q^{-1}} + K_d \frac{2}{T_s} \frac{1-q^{-1}}{3-q^{-1}}, \tag{16}$$

with the left inverse of the reference model parametrized as in (10), with $n_p^r = 2$, and $n_p^c = n_z^r = n_z^c = 0$. In the cost, the barrier function is defined as in (13c), with $k = 10$ and the weights are chosen as $W_e = 0.01$, $W_{\Delta u} = 100$ and $W_u = 0.5$. Algorithm 1 is run with $N_p = 10$ and $i_{max} = 200$. As in Caré et al. (2019), a first order low-pass filter $W(s)$ with time constant 0.3 is further used to penalize the mismatch between the desired and the actual closed-loop performance at frequencies below 2 [rad/s].

The performance attained by considering a square wave set point is reported in Figure 3, showing that the combined tuning of M^{-1} and the design of the controller with VRFT allows us to satisfactorily track the considered reference. As shown in Figure 4, we are actually able to outperform the result obtained by arbitrarily fixing the reference model as

$$M(s) = \frac{1}{(1+0.1s)(1+0.7s)}, \tag{17}$$

despite the quite performing reference model that has been chosen. Finally, we compare the performance resulting from the use of the proposed technique with the one achieved with the method in

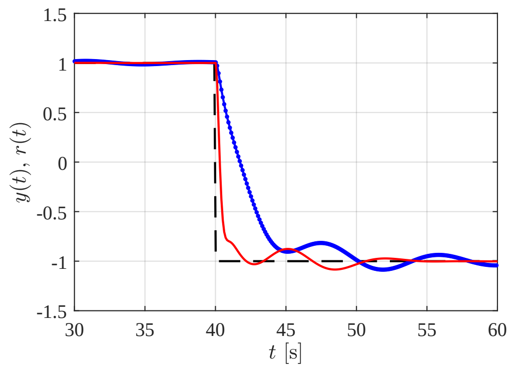

Figure 5: Reference signal (dashed black) vs attained (noiseless) closed-loop outputs obtained data-driven model reference tuning+controller design procedures (detail). Proposed left inverse optimization (red), approach described in Selvi et al. (2018) (dashed dotted blue).

Table 2: RMSE_r (18) vs reference model

	Fixed reference (17)	Approach in Selvi et al. (2018)	Our method
RMSE_r	0.20	0.28	0.15

Selvi et al. (2018). In this second case, we fix the reference r as a square wave, under the assumption that the final user is particularly interested in tracking step references. As shown in Figure 5, the controller resulting from applying our approach allows one to attain better performance than the one obtained with Selvi et al. (2018) when considering a square wave set point, despite the second controller is specifically trained on this class of references. These results are further confirmed by the values of the *Root Mean Square Error* (RMSE), namely

$$\text{RMSE}_r = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(r(t) - y(t))^2}, \tag{18}$$

that are reported in Table 2. The same quality index is used to evaluate the performance of the proposed with respect to two alternative fixed reference models, namely

$$M_2(s) = \frac{1}{(1+0.3s)(1+0.1s)}, \quad M_3(s) = \frac{1}{(1+0.5s)(1+0.05s)}.$$

The values of RMSE_r obtained by considering these prefixed reference models are 0.15 and 0.17 [m], by respectively considering M_2 and M_3, showing that the proposed approach allows us to obtain either the same or better tracking performance than to the one attained by carefully fixing the reference model a priori.

5. Conclusions

In this paper, we have proposed an extension of the VRFT method to tune the left inverse of the reference model as a hyper-parameter, while jointly designing the feedback controller. Particle swarm optimization has been used as an effective numerical tool for this purpose. The advantages over existing data-driven methods have been shown on a benchmark simulation case study. Future work includes the consideration of different cost functions as well as optimal experiment design.

References

A. Bemporad, N.L. Ricker, and J.G. Owen. Model predictive control - new tools for design and evaluation. In *Proceedings of the 2004 American Control Conference*, volume 6, pages 5622–5627 vol.6, June 2004.

M.C. Campi, A. Lecchini, and S.M. Savaresi. Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38(8):1337 – 1346, 2002.

A. Caré, F. Torricelli, M.C. Campi, and S.M. Savaresi. A toolbox for virtual reference feedback tuning (VRFT). In *2019 18th European Control Conference (ECC)*, pages 4252–4257, June 2019.

S. Formentin, K. van Heusden, and A. Karimi. A comparison of model-based and data-driven controller tuning. *International Journal of Adaptive Control and Signal Processing*, 28(10):882–897, 2014.

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

S. Formentin, M.C. Campi, A. Carè, and S.M. Savaresi. Deterministic continuous-time virtual reference feedback tuning (vrft) with application to pid design. *Systems & Control Letters*, 127: 25–34, 2019.

P. Kergus, C. Poussot-Vassal, F. Demourant, and S. Formentin. Frequency-domain data-driven control design in the loewner framework. *IFAC-PapersOnLine*, 50(1):2095–2100, 2017.

P. Kergus, S. Formentin, C. Poussot-Vassal, and F. Demourant. Data-driven control design in the loewner framework: Dealing with stability and noise. In *2018 European Control Conference (ECC)*, pages 1704–1709. IEEE, 2018.

P. Kergus, M. Olivi, C. Poussot-Vassal, and F. Demourant. From reference model selection to controller validation: Application to loewner data-driven control. *IEEE Control Systems Letters*, 2019.

H. Nijmeijer and S.M. Savaresi. On approximate model-reference control of siso discrete-time nonlinear systems. *Automatica*, 34(10):1261–1266, 1998.

R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1): 33–57, 2007.

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

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