IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Data-driven model-reference control with closed-loop stability: the output-feedback case

T.O. de Jong, V. Breschi, *Member, IEEE*, M. Schoukens , *Senior Member, IEEE*, and S. Formentin, *Member, IEEE*

***Abstract*— We generalize a recently introduced data-driven approach for model-reference control design with closed-loop stability guarantees to the case of single-input single-output systems with inaccessible state. By considering a dynamic controller with fixed structure and leveraging a data-based description of the closed-loop dynamics, we propose a two-stage strategy for the optimization of the controller's parameters to match the desired closed-loop behavior. By means of a benchmark simulation example, we show the potential of the proposed approach and the impact of a simple strategy to handle noisy measurements.**

***Index Terms*— Data-driven control, output-feedback control, closed-loop stability**

I. Introduction

In the last years, several works have been reconsidering the classical two-stage approach to learning-based control, namely system identification+model-based design. The main reason behind this shift lays in the fact that standard identification approaches often overlook the ultimate use of the identified model, *i.e.,* the design of an effective controller. While efforts have been carried out to include this last information into identification strategies (see, *e.g.,* [1]), data-driven (DD) control techniques are nowadays being credited as viable alternatives to the standard learning-based control paradigm, allowing one to directly exploit data to learn a controller (bypassing explicit identification phases). Existing data-driven approaches range from "traditional" model reference strategies, *e.g.,* the Virtual Reference Feedback Tuning (VRFT) approach [2], to the newly proposed optimal and predictive strategies founded on Willems' fundamental lemma [3] and behavioral theory (see, *e.g.,* [4]–[6]). While closed-loop stability can eventually be guaranteed when considering this last class of approaches, only few direct, model reference methods have been extended to account for closed-loop stability by design. Extensions to existing approaches have been proposed in [7], [8] to guarantee closed-loop stability, with results that are nonetheless asymptotic (*i.e.,* hold for infinite-dimensional datasets) or rather conservative. Instead, the model reference strategy proposed in [9] directly enforces closed-loop stability by design, but it relies on the (unrealistic) assumption of fully measurable state.

This project 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.

T.O. De Jong, V. Breschi and M. Schoukens are with the Department of Electrical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, Netherlands; S. Formentin is with DEIB, Politecnico di Milano, 20133 Milano, Italy.

In this work, we propose a first extension of the data-driven design framework introduced in [9] to the input/output setting. Specifically, we present a novel strategy for the direct design of a fixed structure controller (endowed with integral action) from a batch of input and output data, which explicitly seeks for closed-loop stability and reference model matching while not performing closed-loop experiments at design time. Notably, the shift from the setting of [9] to that considered in this work greatly increases the complexity of the problem to be solved. Indeed, model matching cannot be simply imposed via (strict or relaxed) matrix equalities as in [9], but it can only be searched for by comparing the desired output behavior to that achieved with the designed controller over a finite set of data points. In turn, this results in a non-convex optimization problem, which we tackle with a new (sub-optimal) two-stage strategy, firstly focusing on model matching and then projecting the model reference controller onto the set of stabilizing laws. To counteract the effect of noise, this work leverages a simple averaging strategy, whose effectiveness is analyzed on a benchmark example.

The paper is structured as follows. Section II introduces the main assumptions of our setup and the design problem, which is constructed in Section III. Our two-stage approach to tackle it is presented in Section IV, while the results of its application on a benchmark example are shown in Section V. The paper is ended by final remarks and directions for future work.

Notation: Given $B \in \mathbb{R}^{m\times n}$, we denote its transpose as $B^\top$, its Frobenius norm as $\|B\|_F$, its Moore-Penrose inverse as $B^\dagger$ and (when $m=n$ and it exists) its inverse as B^{-1}. Identity and zero matrices are respectively indicated as I and 0. If a matrix $Q \in \mathbb{R}^{n\times n}$ is positive definite (positive semi-definite), then $Q \succ 0$ ($Q \succeq 0$). Given $\omega \in \mathbb{R}^p$, we indicate with $\|\omega\|$ its 2-norm. Lastly, for $\nu(t) \in \mathbb{R}^m$, we denote as $N_{i,T-j}$ the Hankel matrix

$$N_{i,T-j} = \begin{bmatrix}\nu(i) & \nu(i+1) & \cdots & \nu(T-j)\end{bmatrix} \in \mathbb{R}^{m\times T-j-i}, \quad (1)$$

for $i,j \in \mathbb{N}$, with $i,j \geq 0$ and $i < T-j$.

II. Background

Consider a *linear, time invariant* (LTI), single-input single-output, strictly proper and *controllable* system $\mathcal{S}$, whose behavior is described by a set of *unknown* difference equations. denote its exogenous input at time $t \in \mathbb{N}_0$ as $u(t) \in \mathbb{R}$ and the associated *noiseless* output as $y^{\text{o}}(t) \in \mathbb{R}$. Given an upper-bound $n \in \mathbb{N}$ on the order of $\mathcal{S}$, let us further consider the (*unknown*) extended model

$$y^{\text{o}}(t) = \sum_{i=1}^{n} a_i y^{\text{o}}(t-i) + \sum_{j=1}^{n} b_j u(t-j), \tag{2}$$

describing the input/output behavior of $\mathcal{S}$, and the associated non-minimal state realization $\hat{x}^{\text{o}}(t) \in \mathbb{R}^{2n}$:

$$\hat{x}^{\text{o}}(t) = \left[y^{\text{o}}(t-n) \cdots y^{\text{o}}(t-1) \;\; u(t-n) \cdots u(t-1) \right]^{\top}, \tag{3}$$

comprising n past inputs/outputs. Accordingly, we can easily obtain the following (non-minimal) state-space model

$$\hat{x}^{\text{o}}(t+1) = \underbrace{\begin{bmatrix} 0 & I & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I & 0 & 0 & \cdots & 0 \\ a_n & a_{n-1} & \cdots & a_1 & b_n & b_{n-1} & \cdots & b_1 \\ 0 & 0 & \cdots & 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & I \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathcal{A} \in \mathbb{R}^{2n \times 2n}} \hat{x}^{\text{o}}(t) + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{\mathcal{B} \in \mathbb{R}^{2n}} u(t),$$

$$y^{\text{o}}(t) = \underbrace{\begin{bmatrix} a_n & a_{n-1} & \cdots & a_1 & b_n & b_{n-1} & \cdots & b_1 \end{bmatrix}}_{\mathcal{C} \in \mathbb{R}^{1 \times 2n}} \hat{x}^{\text{o}}(t), \tag{4}$$

which we suppose satisfies the following assumption.

Assumption 1: The $2n$-dimensional state-space model in (4) is controllable.

Remark 1 (On the controllability of (4)): Let

$$A(q^{-1}) = 1 - a_1 q^{-1} - \ldots - a_n q^{-n},$$
$$B(q^{-1}) = b_1 q^{-1} + \ldots + b_n q^{-n},$$

be the polynomials in the back shift operator[1] q^{-1} characterizing (2). Then, the extended state-space model (4) is controllable if they are coprime (see [10, Lemma 3.4.7]).

Despite no model for $\mathcal{S}$ is known, suppose that we have access to a set of input/output pairs $\mathcal{D}_T = \{\mathcal{U}_T, \mathcal{Y}_T\}$ of length

$$T \geq 4n + 2, \tag{5}$$

with $\mathcal{Y}_T$ verifying the following assumption.

Assumption 2: The outputs in $\mathcal{Y}_T = \{y(t)\}_{t=0}^{T-1}$ are corrupted by zero-mean, white noise with covariance $\Omega \in \mathbb{R}$, *i.e.,*

$$y(t) = y^{\text{o}}(t) + v(t). \tag{6}$$

Let us also assume that $\mathcal{U}_T$ satisfies the following.

Assumption 3: The sequence $\mathcal{U}_T = \{u(t)\}_{t=0}^{T-1}$ is persistently exciting of order $2n+1$ according to [3].

Then, the rank condition

$$\text{rank}\left(\begin{bmatrix} U_{n,T-1} \\ \hat{X}_{n,T-1} \end{bmatrix} \right) = 2n + 1, \tag{7}$$

holds, where $U_{n,T-1}$ and $\hat{X}_{n,T-1}$ are Hankel matrices (1) of the inputs and *noisy* extended states. Note that, condition (5) is needed for $\mathcal{U}_T$ to satisfy Assumption 3 (see [4, Section II.A]).

Remark 2 (On (7)): While in a noise-free setting (7) is verified only when n corresponds to the true order of $\mathcal{S}$ (see, *e.g.,* the discussion in [11, Section 3.3]), this is no longer necessary when data are corrupted by noise.

A. Goal

In this work, we aim to use the available batch of data to design a *stabilizing* controller $\mathcal{K}$ (endowed with integral action) to enable the closed-loop output to track as closely as possible the *desired behavior* $y^d(t) \in \mathbb{R}$ dictated by the stable reference model $\mathcal{M}$:

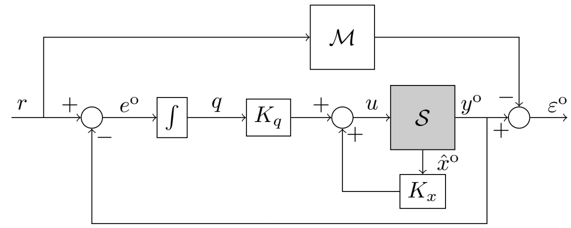


Fig. 1. Ideal matching scheme, where $e^{\text{o}} = r - y^{\text{o}}$, $\varepsilon^{\text{o}} = y^{\text{o}} - y^d$ is the mismatching error and $\hat{x}^{\text{o}}$ is the extended state (see (3)).

$$\mathcal{M}: \quad \begin{cases} x^d(t+1) = A_M x^d(t) + B_M r(t), \\ y^d(t) = C_M x^d(t) + D_M r(t), \end{cases} \tag{8}$$

where $x^d(t) \in \mathbb{R}^{n_d \times n_d}$ indicates the state of the reference model, while $r(t) \in \mathbb{R}$ is any possible user-defined set point. The matrices $A_M \in \mathbb{R}^{n_d \times n_d}$, $B_M \in \mathbb{R}^{n_d \times 1}$, $C_M \in \mathbb{R}^{1 \times n_d}$ and $D_M \in \mathbb{R}$ are assumed to be fixed by the user beforehand, and chosen for $I - A_M$ to be invertible and for

$$C_M (I - A_M)^{-1} B_M + D_M = 1,$$

to guarantee zero steady-state error with respect to step-like set points. Meanwhile, among alternative structures[2], we focus on a controller parameterized as:

$$\mathcal{K}: \quad u(t) = K_x \hat{x}^{\text{o}}(t) + K_q q^{\text{o}}(t), \tag{9a}$$

with

$$q^{\text{o}}(t) = q^{\text{o}}(t-1) + e^{\text{o}}(t) = q^{\text{o}}(t-1) + (r(t) - y^{\text{o}}(t)), \tag{9b}$$

being the *integral* of the tracking error at time t, and $K_x \in \mathbb{R}^{1 \times 2n}$ and $K_q \in \mathbb{R}$ being the gains to be designed. Note that, based on our structural choices, we separately map the effect of past input/output behaviors and tracking performance onto the control action. Meanwhile, due to its dependence on the extended state, $\mathcal{K}$ turns out to be *dynamic* (differently from the static controller considered in [9]).

III. Building the Matching Problem with Stability

Let us consider the dynamics in (4). By further augmenting the state as follows:

$$\chi^{\text{o}}(t) = \left[\hat{x}^{\text{o},\top}(t) \quad q^{\text{o},\top}(t-1) \right]^{\top}. \tag{10}$$

and merging (4) with the input definition (see (9)), the overall closed-loop dynamics is given by

$$\chi^{\text{o}}(t+1) = \underbrace{\begin{bmatrix} \mathcal{A} + \mathcal{B}K & \mathcal{B}K_q \\ -\mathcal{C} & I \end{bmatrix}}_{\mathcal{A}_{cl}(K_x, K_q)} \chi^{\text{o}}(t) + \underbrace{\begin{bmatrix} \mathcal{B}K_q \\ I \end{bmatrix}}_{\mathcal{B}_{cl}(K_q)} r(t) \tag{11a}$$

$$y^{\text{o}}(t) = \begin{bmatrix} \mathcal{C} & 0 \end{bmatrix} \chi^{\text{o}}(t) = \mathcal{C}_{cl} \chi^{\text{o}}(t), \tag{11b}$$

with

$$K = K_x - K_q \mathcal{C}. \tag{11c}$$

Accordingly, for the closed-loop system to be stable in the Lyapunov sense, there should exist a *symmetric*, *positive definite* matrix $P = P^{\top} \in \mathbb{R}^{(2n+1) \times (2n+1)}$, such that:

$$\mathcal{A}_{cl}(K_x, K_q) P \left[\mathcal{A}_{cl}(K_x, K_q) \right]^{\top} - P \prec 0. \tag{12}$$

Based on the matching scheme in Fig. 1, the problem loosely stated in Section II can be formalized as follows:

[1] $q^{-j} u(t) = u(t-j)$, $\forall j \in \mathbb{Z}$.

[2] The integral action can be also enforced by constraining the control gains.

$$\underset{E^{\text{o}},K_x,K_q,P}{\text{minimize}} \quad \|E^{\text{o}}\|^2 \tag{13a}$$

$$\text{s.t. } \varepsilon^{\text{o}}(t) = \mathcal{C}\hat{x}^{\text{o}}(t) - y^d(t), \quad \forall t, \tag{13b}$$

$$u(t) = K_x\hat{x}^{\text{o}}(t) + K_q q^{\text{o}}(t), \quad \forall t, \tag{13c}$$

$$\hat{x}^{\text{o}}(t+1) = \mathcal{A}\hat{x}^{\text{o}}(t) + \mathcal{B}u(t), \quad \forall t, \tag{13d}$$

$$q^{\text{o}}(t) = q^{\text{o}}(t-1) + (r(t) - y^{\text{o}}(t)), \quad \forall t, \tag{13e}$$

$$P \succ 0, \ \mathcal{A}_{cl}(K_x, K_q)P\,[\mathcal{A}_{cl}(K_x, K_q)]^\top - P \prec 0, \tag{13f}$$

where E^{o} indicates the Hankel matrix stacking all the mismatching errors $\varepsilon^{\text{o}}(t), \forall t$, while the dependence of $u(t)$ on the reference in (13c) is made explicit by manipulating (9a). This problem thus seeks for a control law implicitly prioritizing one objective with respect to the other. Indeed, closed-loop stability is enforced by constraint, while mismatching errors are only steered toward zero by the chosen cost (that still allows $\varepsilon^{\text{o}}(t)$ to be eventually non-zero for some t).

Remark 3 (On the feasibility of (13)*):* The constraints in (13c)-(13e) are introduced to define the matching error in (13b) (which is then lifted to the cost) and, thus, they are not critical for the feasibility of (13). Meanwhile, (13f) can be satisfied if and only if the system with extended state (10) is *stabilizable*.

A. Removing the dependence on the reference

Since we aim at matching the behavior of $\mathcal{M}$ for any possible user-defined set point, the dependence on $r(t)$ of (13) might be problematic. Indeed, it would require the user to select a single set point to be tracked beforehand, eventually leading to a controller that matches the desired behavior only for the given reference. Not to undermine the generality of the designed controller, while lifting the burden of choosing the set point from the user, in the same spirit of the VRFT approach [12], we replace $r(t)$ with a *fictitious reference* constructed based on $\mathcal{M}$. To this end, with a slight abuse of notation, let us compactly define the input/output relationship of $\mathcal{M}$ as

$$y^d(t) = \mathcal{M}r(t), \tag{14}$$

according to which (13b) can be recast as:

$$\varepsilon^{\text{o}}(t) = y^{\text{o}}(t) - \mathcal{M}r(t). \tag{15}$$

Additionally, let us assume that the relationship in (14) can be inverted, namely that there exists $\mathcal{M}^\dagger$ such that[3]

$$r(t) = \mathcal{M}^\dagger y^d(t). \tag{16}$$

Exploiting this definition, we can define the *fictitious reference*

$$r^f(\varepsilon^{\text{o}}(t)) = \mathcal{M}^\dagger\left[y^{\text{o}}(t) - \varepsilon^{\text{o}}(t)\right]. \tag{17}$$

In turn, this allows us to rewrite (13c) as follows:

$$u(t) = K_x\hat{x}^{\text{o}}(t) + K_q\sum_{\tau=0}^{t}\tilde{y}^{\text{o}}(\tau) - K_q\sum_{\tau=0}^{t}\mathcal{M}^\dagger\varepsilon^{\text{o}}(\tau), \tag{18a}$$

with $\tilde{y}^{\text{o}}(t) = \mathcal{M}^\dagger y^{\text{o}}(t) - y^{\text{o}}(t)$ and the terms in K_q easily follow from the definition of the integral dynamics and that of the fictitious reference. We can now isolate the term depending on the mismatching error, *i.e.,*

$$\underbrace{K_q\sum_{\tau=0}^{t}\mathcal{M}^\dagger\varepsilon^{\text{o}}(\tau)}_{\epsilon(K_q,\varepsilon^{\text{o}})} = K_x\hat{x}^{\text{o}}(t) + K_q\sum_{\tau=0}^{t}\tilde{y}^{\text{o}}(\tau) - u(t), \tag{18b}$$

[3] $\mathcal{M}^\dagger$ can be obtained as explained in [12, Proposition 1, Section 7].

and replace the matching problem in (13) with

$$\underset{E^{\text{o}},K_x,K_q,P}{\text{minimize}} \quad \|E^{\text{o}}\|_F^2 \tag{19a}$$

$$\text{s.t. } \epsilon(K_q,\varepsilon^{\text{o}}) = K_x\hat{x}^{\text{o}}(t) + K_q\sum_{\tau=0}^{t}\tilde{y}^{\text{o}}(\tau) - u(t), \quad \forall t, \tag{19b}$$

$$\hat{x}^{\text{o}}(t+1) = \mathcal{A}\hat{x}^{\text{o}}(t) + \mathcal{B}u(t), \quad \forall t, \tag{19c}$$

$$\tilde{y}^{\text{o}}(t) = \mathcal{M}^\dagger y^{\text{o}}(t) - y^{\text{o}}(t), \quad \forall t, \tag{19d}$$

$$P \succ 0, \ \mathcal{A}_{cl}(K_x, K_q)P\,[\mathcal{A}_{cl}(K_x, K_q)]^\top - P \prec 0, \tag{19e}$$

Note that, based on the definition of the fictitious reference, (19) is equivalent to (13). Meanwhile, the new problem does not feature the constraints in (13b) and (13e), since they are exploited in the definition of the fictitious reference (see (17)) and the input in (18a), respectively. It is also worth pointing out that the left-hand-side of (19b) depends on the product of K_q and a filtered version of the mismatching error, thus introducing a *non-linearity* in two optimization variables.

B. From model-based to (deterministic) DD matching

Problem (19) still relies on the matrices $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ in (11), which are *unknown* in our setup. By exploiting arguments similar to that of [4], we now replace the model-based formulation with its data-driven counterpart still focusing on a *deterministic* setting, *i.e.,* the output data are *noiseless*. Let us initially consider the dynamics of the non-minimal state $\hat{x}^{\text{o}}(t)$ only. The latter can be equivalently written as:

$$\hat{x}^{\text{o}}(t+1) = [\mathcal{B}\ \mathcal{A}]\left\{\begin{bmatrix}K\\ I\end{bmatrix}\hat{x}^{\text{o}}(t) + \begin{bmatrix}K_q\\ 0\end{bmatrix}(r(t) + q^{\text{o}}(t-1))\right\}. \tag{20}$$

This model-based expression can be translated into its data-driven counterpart as follows:

$$\hat{x}^{\text{o}}(t+1) = \hat{X}^{\text{o}}_{n+1,T}G\hat{x}^{\text{o}}(t) + \hat{X}^{\text{o}}_{n+1,T}G_q(r(t) + q^{\text{o}}(t-1)), \tag{21a}$$

with the additional consistency conditions

$$\hat{X}^{\text{o}}_{n,T-1}G = I, \quad \hat{X}^{\text{o}}_{n,T-1}G_q = 0, \tag{21b}$$

where $G \in \mathbb{R}^{(T-n+1)\times 2n}$, $G_q \in \mathbb{R}^{T-n+1}$, and $\hat{X}_{n+1,T}$ is the Hankel matrix of one-step-ahead extended states. This result is formalized in the following Lemma.

Lemma 1 (Deterministic data-driven extended dynamics): Let the *noiseless* dataset $\mathcal{D}_T^{\text{o}} = \{\mathcal{U}_T, \mathcal{Y}_T^{\text{o}}\}$ be informative and long enough (in the spirit of (7) and (5), respectively). Then, the models in (20) and (21a) are equivalent, provided that G and G_q in (21a) satisfy the consistency conditions in (21b).

Proof: Thanks to Assumptions 3, by the Rouché-Capelli theorem, there exists two matrices $G \in \mathbb{R}^{(T-n+1)\times 2n}$ and $G_q \in \mathbb{R}^{(T-n+1)}$ such that:

$$\begin{bmatrix}K\\ I\end{bmatrix} = \begin{bmatrix}U_{n,T-1}\\ \hat{X}^{\text{o}}_{n,T-1}\end{bmatrix}G, \qquad \begin{bmatrix}K_q\\ 0\end{bmatrix} = \begin{bmatrix}U_{n,T-1}\\ \hat{X}^{\text{o}}_{n,T-1}\end{bmatrix}G_q. \tag{22}$$

Accordingly, the following holds:

$$[\mathcal{B}\ \mathcal{A}]\begin{bmatrix}K\\ I\end{bmatrix} = [\mathcal{B}\ \mathcal{A}]\begin{bmatrix}U_{n,T-1}\\ \hat{X}^{\text{o}}_{n,T-1}\end{bmatrix}G = \hat{X}^{\text{o}}_{1,T}G,$$

$$[\mathcal{B}\ \mathcal{A}]\begin{bmatrix}K_q\\ 0\end{bmatrix} = [\mathcal{B}\ \mathcal{A}]\begin{bmatrix}U_{n,T-1}\\ \hat{X}^{\text{o}}_{n,T-1}\end{bmatrix}G_q = \hat{X}^{\text{o}}_{1,T}G_q,$$

because of (4), ultimately leading to (21). ■

Note that, the control gains K_x and K_q can be straightforwardly extracted from G and G_q as follows:

$$K_x = U_{n,T-1}G - U_{n,T-1}G_q\hat{\mathcal{C}}^{\text{o}}, \tag{23a}$$
$$K_q = U_{n,T-1}G_q. \tag{23b}$$

Meanwhile, along the line of [4, Section VI], $\hat{\mathcal{C}}^{\text{o}}$ is the data-driven estimate of $\mathcal{C}$, *i.e.,*

$$\hat{\mathcal{C}}^{\text{o}} = \mathbf{1}_n\hat{X}^{\text{o}}_{n+1,T}\begin{bmatrix}U_{n,T-1}\\ \hat{X}^{\text{o}}_{n,T-1}\end{bmatrix}^{\dagger}\begin{bmatrix}0\\ I\end{bmatrix}, \tag{24}$$

and $\mathbf{1}_n$ is the n-th block row selector, namely

$$\mathbf{1}_n = [\underbrace{0 \;\; 0 \;\; \cdots \;\; 0}_{n-1 \text{ times}} \;\; I \;\; \underbrace{0 \;\; 0 \;\; \cdots \;\; 0}_{m \text{ times}}].$$

Based on Lemma 1 and (24), the stability condition in (12) can now be translated into its data-driven counterpart

$$\underbrace{\begin{bmatrix}\hat{X}^{\text{o}}_{n+1,T}G & \hat{X}^{\text{o}}_{n+1,T}G_q\\ -\hat{\mathcal{C}}^{\text{o}} & I\end{bmatrix}}_{\mathcal{A}^{\text{o}}_{cl}(G,G_q)}P\begin{bmatrix}\hat{X}^{\text{o}}_{n+1,T}G & \hat{X}^{\text{o}}_{n+1,T}G_q\\ -\hat{\mathcal{C}}^{\text{o}} & I\end{bmatrix}^{\top} - P \prec 0. \tag{25}$$

Therefore, we can cast the (deterministic) data-driven problem with stability guarantees as:

$$\underset{E^{\text{o}},G,G_q,P}{\text{minimize}} \quad \|E^{\text{o}}\|_F^2 \tag{26a}$$
$$\text{s.t. } \epsilon(G_q,\varepsilon^{\text{o}}) = \hat{u}^{\text{o}}(t;G,G_q) - u(t), \quad t \in \mathcal{I}_T \tag{26b}$$
$$\hat{X}^{\text{o}}_{n,T-1}G = I, \quad \hat{X}^{\text{o}}_{n,T-1}G_q = 0, \tag{26c}$$
$$P \succ 0, \; \mathcal{A}_{cl}(G,G_q)P\left[\mathcal{A}_{cl}(G,G_q)\right]^{\top} - P \prec 0, \tag{26d}$$

removing the constraint in (19d), since $\tilde{y}^{\text{o}}(t)$ is computed from data for all $t \in \mathcal{I}_T$. Note that, due to the data-driven nature of the problem, model matching is now sought over the available data only. Therefore, $E^{\text{o}} \in \mathbb{R}^{1\times T-n+1}$ now stacks the *noiseless* matching errors over the dataset $\mathcal{D}^{\text{o}}_T$, $u(t)$ are the input samples available for data-driven design, $\mathcal{I}_T = \{n,\dots,T-1\}$, $\hat{u}^{\text{o}}(t;G,G_q)$ is

$$\hat{u}^{\text{o}}(t;G,G_q) = U_{n,T-1}G\hat{x}^{\text{o}}(t) + U_{n,T-1}G_q\sum_{\tau=0}^{t}\tilde{y}^{\text{o}}(t), \tag{26e}$$

and $\hat{x}^{\text{o}}(t)$ is also computed based on the available data, while

$$\epsilon(G_q,\varepsilon^{\text{o}}) = U_{n,T-1}G_q\sum_{\tau=0}^{t}\mathcal{M}^{\dagger}\varepsilon^{\text{o}}(\tau), \quad \forall t \in \mathcal{I}_T. \tag{26f}$$

Thanks to the equivalence established by Lemma 1, problem (26) corresponds to a simplified version of (19), where matching is required only over a finite set of instants. Therefore, (26) admits a solution whenever (19) is feasible.

C. Handling noisy data

Differently from before, we now approach the design problem in a realistic (namely *noisy*) setting, based on Assumption 2. This shift from a noise-free to a noisy setup entails that the equivalence in Lemma 1 does not hold anymore and, thus, matching and/or stability might not be attained by simply replacing noisy data into (26). In this work, we cope with this issue as in [9], [13], by trying to recover the noiseless data from the noisy ones based on the following assumption.

Assumption 4: Multiple data collection experiments can be performed using the same input sequence $\mathcal{U}_T$, leading to $N \geq 1$ datasets $\mathcal{D}^h_T = \{\mathcal{U}_T, \mathcal{Y}^h_T\}$, with $h = 1,\dots,N$.

Accordingly, we can define the averaged extended state

$$\hat{x}^{\text{av}}(t) = \left[\bar{y}(t-n) \cdots \bar{y}(t-1) \;\; u(t-n) \cdots u(t-1)\right]^{\top}, \tag{27a}$$

where $\bar{y}(t)$ is the average of the N available outputs, *i.e.,*

$$\bar{y}(t) = \frac{1}{N}\sum_{h=1}^{N}y^h(t). \tag{27b}$$

In turn, this implies that $\hat{x}^{\text{av}}(t)$ asymptotically (namely for $N \to \infty$) corresponds to $\hat{x}^{\text{o}}(t)$ almost surely, thanks to the features of the measurement noise[4] (see Assumption 2). By replacing the Hankel matrices $\hat{X}^{\text{o}}_{n,T-1}$ and $\hat{X}^{\text{o}}_{n+1,T}$ with those comprising the averaged extended states[5], namely $\hat{X}^{\text{av}}_{n,T-1}$ and $\hat{X}^{\text{av}}_{n+1,T}$, we can ultimately recast the problem as follows:

$$\underset{E,G,G_q,P}{\text{minimize}} \quad \|E\|_F^2 \tag{28a}$$
$$\text{s.t. } \epsilon(G_q,\varepsilon) = \hat{u}^{\text{av}}(t;G,G_q) - u(t), \quad t \in \mathcal{I}_T \tag{28b}$$
$$\hat{X}^{\text{av}}_{n,T-1}G = I, \quad \hat{X}^{\text{av}}_{n,T-1}G_q = 0, \tag{28c}$$
$$P \succ 0, \; \mathcal{A}^{\text{av}}_{cl}(G,G_q)P\left[\mathcal{A}^{\text{av}}_{cl}(G,G_q)\right]^{\top} - P \prec 0, \tag{28d}$$

where E being the collection of matching errors resulting from the averaged dataset,

$$\mathcal{A}^{\text{av}}_{cl}(G,G_q) = \begin{bmatrix}\hat{X}^{\text{av}}_{n+1,T}G & \hat{X}^{\text{av}}_{n+1,T}G_q\\ -\hat{\mathcal{C}}^{\text{av}} & I\end{bmatrix}, \tag{28e}$$

$$\hat{u}^{\text{av}}(t;G,G_q) = U_{n,T-1}G\hat{x}^{\text{av}}(t) + U_{n,T-1}G_q\sum_{\tau=0}^{t}\tilde{y}^{\text{av}}(\tau), \tag{28f}$$

and

$$\hat{\mathcal{C}}^{\text{av}} = \mathbf{1}_n\hat{X}^{\text{av}}_{1,T}\begin{bmatrix}U_{n,T-1}\\ \hat{X}^{\text{av}}_{n,T-1}\end{bmatrix}^{\dagger}\begin{bmatrix}0\\ I\end{bmatrix}, \quad \tilde{y}^{\text{av}}(t) = \mathcal{M}^{\dagger}\bar{y}(t) - \bar{y}(t).$$

Note that, when N tends to infinity, one would expect to recover (26). In this scenario, (28) should thus be feasible if that (deterministic) data-based problem admits a solution.

IV. A Two-Stage LMI-Based Approach for Model Matching with Stability

All constraints in problem (28) feature products between optimization variables. Nonetheless, standard techniques [14] can be exploited to turn (28d) into a *linear matrix inequality* (LMI). Indeed, by recasting $\mathcal{A}_{cl}(G,G_q)$ as

$$\mathcal{A}_{cl}(G,G_q) = \underbrace{\begin{bmatrix}\hat{X}^{\text{av}}_{n+1,T} & 0 & 0\\ 0 & -\hat{\mathcal{C}}^{\text{av}} & I\end{bmatrix}}_{\tilde{\mathcal{A}}_{cl}}\underbrace{\begin{bmatrix}G & G_q\\ I & 0\\ 0 & I\end{bmatrix}}_{\tilde{G}}, \tag{29}$$

the second inequality in (28d) becomes

$$\tilde{\mathcal{A}}_{cl}\tilde{G}P\tilde{G}^{\top}(\tilde{\mathcal{A}}_{cl})^{\top} - P \geq 0. \tag{30a}$$

that can be equivalently rewritten as

$$\begin{bmatrix}P & \tilde{\mathcal{A}}_{cl}Q\\ (\tilde{\mathcal{A}}_{cl}Q)^{\top} & P\end{bmatrix} \succ 0, \tag{30b}$$

after defining $Q = \tilde{G}P^{-1}$ and a Schur complement. However, introducing Q leads to the following translation of (28b):

[4]This result can be easily proven based on the law of large numbers.

[5]They are simply obtained by replacing y^{o} with $\bar{y}$ in their definitions.

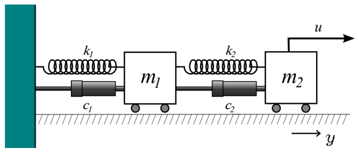


Fig. 2. Scheme of the controlled system [15]. The input is the force applied to $\boldsymbol{m_2}$, while $\boldsymbol{y}$ is the position of $\boldsymbol{m_2}$.

$$\epsilon(G_q,\varepsilon)=\begin{bmatrix}U_{n,T-1} & 0 & 0\end{bmatrix}\underbrace{QP^{-1}}_{\tilde{G}}\begin{bmatrix}\hat{x}^{\mathrm{av}}(t)\\ \sum_{\tau=0}^{t}\tilde{y}^{\mathrm{av}}(\tau)\end{bmatrix},$$

that ultimately becomes less tractable than before, due to its dependence on P^{-1}. To overcome this limitation, we tackle (28) with a sub-optimal two-stage approach, by firstly finding the matching controller and then projecting it onto the set of stabilizing laws (as summarized in Algorithm 1). Specifically, we initially disregard the stability and consistency constraints in (28), considering the "reduced" matching problem:

$$\underset{E,G,G_q}{\text{minimize}} \quad \|E\|_F^2 \tag{31a}$$

$$\text{s.t. } \epsilon(G_q,\varepsilon)=\hat{u}^{\mathrm{av}}(t;G,G_q)-u(t), \quad t\in\mathcal{I}_T. \tag{31b}$$

Note that the consistency constraint in (28c) is not considered, with this problem being solely devoted to design a control action parameterized as in (28f) to attain model matching. Despite this simplification, it still features a non-linear constraints (see the left-hand-side of (31b)). In line with [12], we avoid this issue by minimizing the filtered mismatching error, *i.e.*, shifting to the approximated problem (step 1)

$$\underset{G,G_q\neq 0}{\text{minimize}} \quad \underbrace{\sum_{t=n}^{T-1}\|u(t)-\hat{u}^{\mathrm{av}}(t;G,G_q)\|_2^2}_{J(G,G_q)}. \tag{32}$$

It is worth remarking that in solving (32) we have removed the trivial solution $G_q=0$ from the feasible set, for $\epsilon(G_q,\varepsilon)$ not to be steered to zero by G_q becoming zero itself. By solving this problem, we get a first estimate of the optimization variables G^* and G_q^*, which can be used to construct $\tilde{G}^*$ according to (29). Then (step 3), we seek closed-loop stability by solving

$$\underset{P,Q}{\text{minimize}} \quad \|\tilde{G}^*P-Q\|_F, \tag{33a}$$

$$\text{s.t. } P\succ 0, \quad \begin{bmatrix}P & \tilde{\mathcal{A}}_{cl}Q\\ (\tilde{\mathcal{A}}_{cl}Q)^\top & P\end{bmatrix}\succ 0, \tag{33b}$$

$$\begin{bmatrix}\hat{X}_{n,T-1}^{\mathrm{av}} & 0 & 0\end{bmatrix}Q=\begin{bmatrix}I & 0\end{bmatrix}P, \tag{33c}$$

namely searching for a stabilizing law that resembles the one obtained at the first step, while satisfying the consistency constraints. From the resulting matrices $Q^\star$ and $P^\star$, we can then retrieve $\tilde{G}^\star=Q^\star(P^\star)^{-1}$ and consequently extract the gains of our controller (step 6). Problem (33) is feasible when the system with extended state composed by $\hat{x}^{\mathrm{av}}(t)$ and the integral state is stabilizable. We stress again that solving (32)+(33) is not equivalent to tackling (28), so the obtained controller is sub-optimal and might lead to conservatism.

V. Numerical results

The effectiveness of the proposed approach is assessed on the benchmark system schematized in Fig. 2 [15, Section

Algorithm 1 Data-driven matching with stability

Input: Dataset $\overline{\mathcal{D}}_T=\{\mathcal{U}_T,\bar{\mathcal{Y}}_T\}$, with $\overline{\mathcal{Y}}_T=\{\bar{y}(t)\}_{t=0}^{T-1}$.

1. **find** $G^*,G_q^*\leftarrow\arg\min_{G,G_q\neq 0} J(G,G_q)$;
2. **construct** $\tilde{G}^*$ as in (29), with $G=G^*$ and $G_q=G_q^*$;
3. **find** $P^\star,Q^\star\leftarrow\arg\min_{Q,P}$ s.t. (33b)–(33c) $\|\tilde{G}^*P-Q\|_F$;
4. **compute** $\tilde{G}^\star=Q^\star(P^\star)^{-1}$;
5. **extract** $G^\star$ and $G_q^\star$ from $\tilde{G}^\star$ (see (29));
6. **retrieve** $K_x^\star$ and $K_q^\star$ from $G^\star$ and $G_q^\star$ as in (23);

Output: Estimated gains $K_x^\star$ and $K_q^\star$.

TABLE I
MAIN PARAMETERS OF THE BENCHMARK SYSTEM (REFER TO FIG. 2)

Parameter	m_1 [kg]	m_2 [kg]	c_1 [kg/s]	c_2 [kg/s]	k_1 [kg/s^2]	k_2 [kg/s^2]
Value	1	0.5	0.2	0.5	1	0.5

IV.A], with the parameters in Table I. The reference model is described by the transfer function in the Laplace domain:

$$\mathcal{M}: \quad Y^d(s)=\frac{1}{3s+1}R(s), \tag{34}$$

where $Y^d(s)$ and $R(s)$ are the Laplace transforms of the desired output and reference position, respectively. Toward the design of $\mathcal{K}$ in (9a), we fix $n=4$ (see (3)), thus verifying Assumption 1 while assuming to know the exact order of the system. Meanwhile, to satisfy Assumption 3, we collect $N=100$ realizations of the data with the same pseudo-binary random open-loop input varying in $[-1,1]$ for 10.4 s ($T=104$, as the sampling time is $T_s=0.1$ s), with outputs corrupted by a zero-mean, white noise with standard deviation 0.05, yielding an average signal-to-noise ratio (SNR) of 21.7 [dB].

To assess the robustness of our approach to different realizations of the noise in the training data, we perform 100 Monte Carlo data collection for each of the $N=100$ repeated experiments, ultimately obtaining 100 averaged datasets that are used to train different controllers with Algorithm 1. As evidenced by the *noise-free* trajectories[6] obtained by closing the loop with the designed $\mathcal{K}$ in Fig. 3, the closed-loop behavior of the system is not heavily impacted by the differences in the controllers learned via different realizations of the training set. This result can be related to the use of the noise handling strategy proposed in Section III-C, as highlighted by the results in Table II. Indeed, by progressively increasing N, closed-loop performance tend to gradually improve[7] as expected, since the experimental conditions get progressively "closer" to the asymptotic ones. Meanwhile, by looking at the result in Table III, tracking performance seems to counter-intuitively improve when the noise level increases. Nonetheless, this slight improvement comes at a price of an increase in the output oscillations as shown by the tracking errors in Fig. 4.

We further juxtapose the attained performance with those achieved by closing the loop with two controllers tuned with the VRFT approach, by using the VRFT Toolbox with the same specifications used in [15]. In this comparison, we

[6]The optimization problems are solved with the CVX package [16], [17].

[7]RMSE denotes the average root mean square error between the attained closed-loop output and the set point.

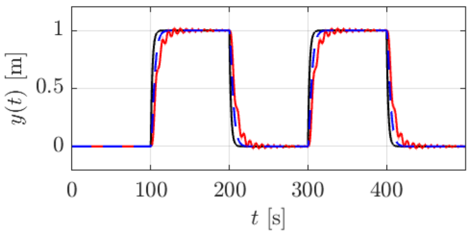


Fig. 3. Mean (red line) and standard deviation (shaded area) of the noiseless closed-loop response over the 100 Monte Carlo averaged datasets *vs* reference position to be tracked (black line), desired behavior (blue dashed line). Note that the shaded area is barely visible.

TABLE II
PERFORMANCE INDEXES *VS* NUMBER N OF REPEATED EXPERIMENTS OVER 100 MONTE CARLO DATASETS.

N	1	10	100	1000
n. diverging instances [%]	6	0	0	0
$\overline{\textbf{RMSE}}$ [m]	0.24	0.16	0.15	0.13

TABLE III
PERFORMANCE INDEXES *VS* AVERAGE SNR OVER 100 MONTE CARLO DATASETS.

$\overline{\textbf{SNR}}$ [dB]	236	156	76	56	22
n. diverging instances [%]	0	0	0	0	0
$\overline{\textbf{RMSE}}$ [m]	0.17	0.17	0.17	0.16	0.15

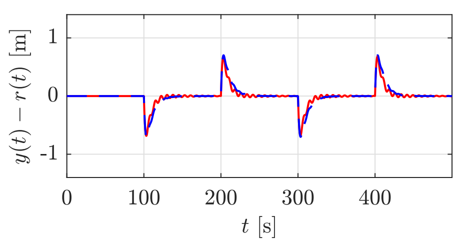
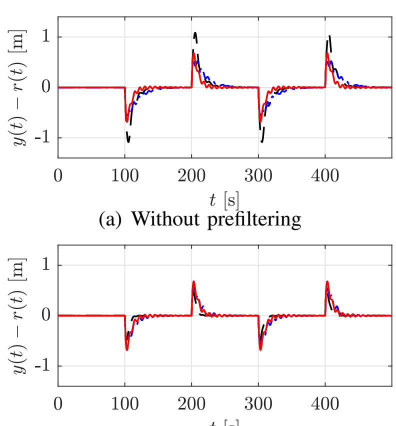


Fig. 4. Mean noiseless tracking error over 100 Monte Carlo average datasets for SNR$=\infty$ [dB] (dashed blue) and SNR$=21.7$ [dB] (red).



(a) Without prefiltering

(b) With prefiltering

Fig. 5. Mean noiseless closed-loop tracking error over 100 Monte Carlo averaged datasets for our controller (red), the PI (dashed blue) and I-FIR (dashed dotted black).

consider a proportional, integral (PI) controller with integral action parameterized as in [15], and a 9-th order finite impulse response controller with integral action (I-FIR). This last controller is chosen since it features the same amount of tunable parameters of the one we propose (see (9a)). Meanwhile, we design these controllers with and without employing the bias shaping filter used in VRFT (see [2] for additional details), since only in this second scenario the comparison with VRFT is truly fair. As shown in Fig. 5, the proposed controller tends to outperform both the I-FIR and the PI controller. Meanwhile, the introduction of the pre-filter visibly improves the performance of both the PI and I-FIR, with the proposed controller still performing comparably to the PI, and the I-FIR controller results in a slightly faster response.

VI. CONCLUSIONS

We presented a two-stage approach to design a class of stabilizing output feedback controllers with endowed integral action in a model reference setup. The results attained with the method on a benchmark case study show its potential and the effectiveness of our simple noise handling strategy.

Future works will focus on a formal analysis on the feasibility, robustness and sub-optimality of the proposed two-stage approach, the introduction of alternative strategies for noise handling in the considered finite sample scenario, and the generalization to other controller classes and to the multi-input multi-output setting.

REFERENCES

[1] S. Formentin and A. Chiuso. Control-oriented regularization for linear system identification. *Automatica*, 127:109539, 2021.
[2] 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.
[3] J.C. Willems, P. Rapisarda, I. Markovsky, and B.L.M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.
[4] C. De Persis and P. Tesi. Formulas for data-driven control: stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924, 2019.
[5] J. Berberich, J. Köhler, M.A. Müller, and F. Allgöwer. Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 66(4):1702–1717, 2021.
[6] J. Coulson, J. Lygeros, and F. Dörfler. Data-enabled predictive control: In the shallows of the DeePC. In *18th European Control Conference (ECC)*, pages 307–312, June 2019.
[7] D. Selvi, D. Piga, G. Battistelli, and A. Bemporad. Optimal direct data-driven control with stability guarantees. *European Journal of Control*, 59:175–187, 2021.
[8] 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.
[9] V. Breschi, C. De Persis, S. Formentin, and P. Tesi. Direct data-driven model-reference control with Lyapunov stability guarantees. In *60th IEEE Conference on Decision and Control*, pages 1456–1461, 2021.
[10] G.C. Goodwin and K.S. Sin. *Adaptive filtering prediction and control*. Courier Corporation, 2014.
[11] J. Berberich, J. Köhler, M.A. Müller, and F. Allgöwer. On the design of terminal ingredients for data-driven MPC. *IFAC-PapersOnLine*, 54(6):257–263, 2021.
[12] S. Formentin, D. Piga, R. Tóth, and S. M. Savaresi. Direct learning of LPV controllers from data. *Automatica*, 65:98–110, 2016.
[13] A. Sassella, V. Breschi, and S. Formentin. Data-driven design of explicit predictive controllers. In *IEEE 61st Conference on Decision and Control*, pages 2821–2826, 2022.
[14] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM studies in applied mathematics: 15, 1994.
[15] A. Carè, F. Torricelli, M.C. Campi, and S.M. Savaresi. A toolbox for Virtual Reference Feedback Tuning (VRFT). In *60th IEEE Conference on Decision and Control*, pages 1456–1461, 2021.
[16] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.
[17] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008.