Data-Driven Inversion-Based Control of Nonlinear Systems with Guaranteed Closed-Loop Stability

Data-driven inversion-based control of nonlinear systems with guaranteed closed-loop stability

Carlo Novara, *Senior Member, IEEE*, and Simone Formentin, *Member, IEEE*

***Abstract*—The Data-Driven Inversion-Based Control (D^2-IBC) approach is a recently introduced control design method for uncertain nonlinear systems, relying on a two degree-of-freedom architecture, with a nonlinear controller and a linear controller running in parallel. Unlike other approaches for nonlinear control, D^2-IBC does not require an accurate modeling of the plant and is computationally efficient. In this note, we introduce a finite-gain stability sufficient condition for the closed-loop system and prove that such a condition theoretically holds when a suitable constraint is enforced during the controller design. Finally, we compare the original and the modified methods on a benchmark simulation example, regarding control of the Duffing oscillator.**

I. Introduction

Classical approaches to nonlinear control may fail when the model of the system used for control design is highly uncertain. The methods proposed in the literature to deal with uncertainty in nonlinear control are many, e.g. approximate linearization via feedback [8], feedforward linearization [9], robust feedback linearization [13], sliding mode control [25], model predictive control [14], and data-based approaches like identification for control [6] and direct data-driven control [19]. See also the important books [5], [24] on nonlinear robust control design.

In all applications where data can be easily collected and deriving a physical model of the plant is costly, difficult and/or time-consuming, data-based approaches are usually preferred.

Unfortunately, the theory behind these methods is not yet mature and only a few contributions about data-based approaches have focused on nonlinear systems. Among these, neural networks [27], [23], Virtual Reference Feedback Tuning (VRFT) [2], Direct Feedback (DFK [19]) and the recently introduced Data-Driven Inversion-Based Control (D^2-IBC) [20].

In particular, the latter approach has showed to be effective [3], [20] in that it relies on an architecture composed by a nonlinear controller and a linear controller in parallel, allowing both compensation of nonlinearities and performance boosting. The nonlinear control block is designed according to the Nonlinear Inversion Control (NIC) method [22], while an adapted version of VRFT [1] is employed for the linear part.

Despite many different approaches for joint design of identification and control have been already proposed (see [10] and the reference therein for a comprehensive overview), as far as we are aware, D^2-IBC is the first "identification for control" method for nonlinear dynamical systems, where also stability guarantees are provided and enforced directly in the identification algorithm. Note that D^2-IBC is based on a significantly different philosophy with respect to DFK: in DFK, an inverse model of the plant to control is directly identified from data, and then this inverse model is used as the controller. In D^2-IBC, a model of the plant is identified from data (according to an "identification for control" approach), and then the controller is obtained by on-line inversion of this model. The main advantage of D^2-IBC with respect to DFK is that it allows the control of systems described by a function non-invertible with respect to the command input. Moreover, D^2-IBC is suitable for output feedback, while DFK requires the availability of the full state vector.

Although the D^2-IBC method has proven to be effective in many situations as compared to other nonlinear approaches [3], [20], no theoretical analysis has been carried out so far about closed-loop stability.

In this paper, we introduce a sufficient condition for finite-gain stability of the closed-loop system. Moreover, we show that such a condition can be formulated as a pure data-based constraint for the model identification task. Given the above result, we modify the identification part of the D^2-IBC method such that (i) the controller computed from the identified model is guaranteed to stabilize the closed-loop system (for a sufficiently large number of data), (ii) the modified optimization problem remains convex. As a side result, we numerically show on the benchmark example of [20] that the additional stability constraint only marginally affects the control performance, whereas it is important to stabilize the system in critical conditions. Notice that the result we provide here is valid for Single Input Single Output (SISO) systems only, while a multivariable extension will be the subject of future research.

The remainder of the paper is as follows. The D^2-IBC approach is recalled in Section II. The closed-loop stability analysis is reported in III, while the way the D^2-IBC approach can be modified to embed the stability constraint is discussed in Section IV. Section V illustrates the numerical performance of the proposed approach. Some concluding remarks end the paper.

Notation. A column vector $x \in \mathbb{R}^{n_x \times 1}$ is denoted as $x = (x_1, \ldots, x_{n_x})$. A row vector $x \in \mathbb{R}^{1 \times n_x}$ is denoted as $x = [x_1, \ldots, x_{n_x}] = (x_1, \ldots, x_{n_x})^\top$, where $\top$ indicates the transpose.

A discrete-time signal (i.e. a sequence of vectors) is denoted with the bold style: $\boldsymbol{x} = (x_1, x_2, \ldots)$, where $x_t \in \mathbb{R}^{n_x \times 1}$ and $t = 1, 2, \ldots$ indicates the discrete time; $x_{i,t}$ is the ith component of the signal $\boldsymbol{x}$ at time t.

A regressor, i.e. a vector that, at time t, contains n present

C. Novara is with Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Corso Duca degli Abruzzi, 24 - 10129 Torino, Italy. Email to: carlo.novara@polito.it.

S. Formentin is with Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, via G. Ponzio 34/5, 20133 Milano (Italy). Email to: simone.formentin@polimi.it.

and past values of a variable, is indicated with the bold style and the time index: $\boldsymbol{x}_t = (x_t, \dots, x_{t-n+1})$.
The ℓ_p norms of a vector $x = (x_1, \dots, x_{n_x})$ are defined as

$$\|x\|_p \doteq \begin{cases} \left(\sum_{i=1}^{n_x} |x_i|^p\right)^{\frac{1}{p}}, & p < \infty, \\ \max_i |x_i|, & p = \infty. \end{cases}$$

The ℓ_p norms of a signal $\boldsymbol{x} = (x_1, x_2, \dots)$ are defined as

$$\|\boldsymbol{x}\|_p \doteq \begin{cases} \left(\sum_{t=1}^{\infty} \sum_{i=1}^{n_x} |x_{i,t}|^p\right)^{\frac{1}{p}}, & p < \infty, \\ \max_{i,t} |x_{i,t}|, & p = \infty, \end{cases}$$

where $x_{i,t}$ is the ith component of the signal $\boldsymbol{x}$ at time t.

II. The D^2-IBC approach

Consider a nonlinear discrete-time SISO system in regression form:

$$\begin{aligned} y_{t+1} &= g\left(\boldsymbol{y}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_t\right) \\ \boldsymbol{y}_t &= (y_t, \dots, y_{t-n+1}) \\ \boldsymbol{u}_t &= (u_t, \dots, u_{t-n+1}) \\ \boldsymbol{\xi}_t &= (\xi_t, \dots, \xi_{t-n+1}) \end{aligned} \tag{1}$$

where $u_t \in U \doteq [\underline{u}, \overline{u}] \subset \mathbb{R}$ is the saturated input, $y_t \in \mathbb{R}$ is the output, $\xi_t \in \Xi \doteq \left[-\bar{\xi}, \bar{\xi}\right]^{n_\xi} \subset \mathbb{R}^{n_\xi}$ is a disturbance including both process and measurement noises, and n is the system order. Both U and Ξ are compact sets.

Suppose that the system (1) is unknown, but a set of measurements is available:

$$\mathcal{D} \doteq \{\tilde{u}_t, \tilde{y}_t\}_{t=1-L}^{0} \tag{2}$$

where $\tilde{u}_t \in U$, $\tilde{y}_t \in Y$, $Y = [-\bar{y}, \bar{y}]$ and $\bar{y} \doteq \max_t |\tilde{y}_t| < \infty$. The tilde is used to indicate the input and output samples of the data set (2), which is supposed to be available at time $t = 0$ when the controller needs to be designed. The input signals employed to generate (2) are also assumed to be such that the system output does not diverge.

Let $R \doteq [-\bar{r}, \bar{r}]$, with $0 \leq \bar{r} \leq \bar{y}$, be a domain of interest for the trajectories of the system (1). The aim is to control the system (1) in such a way that, starting from any initial condition $\boldsymbol{y}_0 \in \mathcal{Y}^0 \doteq R^n \subset \mathbb{R}^n$, the system output sequence $\boldsymbol{y} = (y_1, y_2, \dots)$ tracks any reference sequence $\boldsymbol{r} = (r_1, r_2, \dots) \in \mathcal{R} \subseteq R^\infty \subset \ell_\infty$. The set of all possible disturbance sequences is defined as $\Xi \doteq \{\boldsymbol{\xi} = (\xi_1, \xi_2, \dots) : \xi_t \in \Xi, \forall t\}$.

To accomplish this task, we consider the feedback control structure depicted in Figure 1, where S is the system (1) (including also the input saturation), K^{nl} is a nonlinear controller used to guide the system (1) along the trajectories of interest, while K^{lin} is a linear controller aimed to enhance the tracking precision.

To design the nonlinear controller, the first step in D^2-IBC is to identify from the data (2) a model for the system (1) of the form

$$\begin{aligned} \hat{y}_{t+1} &= f\left(\boldsymbol{y}_t, \boldsymbol{u}_t\right) \equiv f\left(\boldsymbol{q}_t, u_t\right) \\ \boldsymbol{q}_t &= (y_t, \dots, y_{t-n+1}, u_{t-1}, \dots, u_{t-n+1}) \end{aligned} \tag{3}$$

where u_t and y_t are the system input and output, and $\hat{y}_t$ is the model output. No assumptions are required on the order of the true system (1). The model order is automatically chosen by the identification algorithm presented in Section IV. Here, for simplicity of notation, the model is chosen of the same order as the system and a one-step prediction horizon is considered. However, the methodology and results presented in the paper can be easily extended to the case where the model and system orders are different and/or a multi-step prediction horizon is adopted (using a prediction horizon larger than 1 may enhance the control performance/robustness).

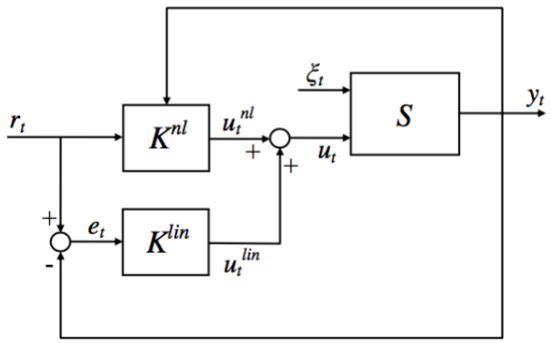

Fig. 1. Feedback control system.

A parametric structure is taken for the function f:

$$f\left(\boldsymbol{q}_t, u_t\right) = \sum_{i=1}^{N} \alpha_i \phi_i\left(\boldsymbol{q}_t, u_t\right) \tag{4}$$

where ϕ_i are polynomial basis functions and α_i are parameters to be identified.

Once a model of the form (3) has been identified, the controller K^{nl} is obtained by its inversion as explained next. Suppose that, at a time $t \geq 0$, the reference value for the time $t+1$ is r_{t+1} and the current regressor is $\boldsymbol{q}_t$. Inversion consists in finding a command input u_t^{nl} such that the model output at time $t+1$ is "close" to r_{t+1}:

$$\hat{y}_{t+1} = f\left(\boldsymbol{q}_t, u_t^{nl}\right) \cong r_{t+1}. \tag{5}$$

The latter equality may be not exact for two reasons: 1) no $u_t^{nl} \in U$ may exist for which $\hat{y}_{t+1}$ is exactly equal to r_{t+1}; 2) values of u_t^{nl} with a limited ℓ_2 norm may be of interest, in order to limit the command activity. This kind of inversion is called (approximate) *right-inversion* and can be performed also when f is not injective with respect to u_t (e.g., for some r_{t+1} and $\boldsymbol{q}_t$, more than one value of u_t may exist such that (5) holds).

The command input u_t^{nl} yielding (5) is found solving the following optimization problem:

$$u_t^{nl} = \arg\min_{\mathfrak{u} \in U} J\left(\mathfrak{u}\right) \tag{6}$$

where

$$J\left(\mathfrak{u}\right) = \frac{1}{\rho_y}\left(r_{t+1} - f\left(\boldsymbol{q}_t, \mathfrak{u}\right)\right)^2 + \frac{\mu}{\rho_u}\mathfrak{u}^2; \tag{7}$$

$\rho_y \doteq \|(\tilde{y}_{1-L}, \dots, \tilde{y}_0)\|_2^2 / L$ and $\rho_u \doteq \|(\tilde{u}_{1-L}, \dots, \tilde{u}_0)\|_2^2 / L$ are normalization constants computed from the data set (2), and $\mu \geq 0$ is a design parameter, determining the trade-off between tracking precision and command activity.

Since a polynomial basis function expansion has been considered for f, the objective function $J(\mathfrak{u})$ is also polynomial in $\mathfrak{u}$. The minima of $J(\mathfrak{u})$ can thus be found by simply considering the roots of its derivative. Define the set

$$U^s \doteq \left(\text{Rroots}\left(\frac{dJ(\mathfrak{u})}{d\mathfrak{u}} \right) \cap U \right) \cup \{\underline{u}, \overline{u}\}$$

where $\text{Rroots}(\cdot)$ denotes the set of all real roots of $\cdot$, and $\underline{u}$ and $\overline{u}$ are the boundaries of U. The optimal command input is given by

$$u_t^{nl} = \arg\min_{\mathfrak{u} \in U^s} J(\mathfrak{u}). \tag{8}$$

Notice that the derivative $dJ(\mathfrak{u})/d\mathfrak{u}$ can be computed analytically. Moreover, U^s is composed by a "small" number of elements. In fact $\text{card}(U^s) < \deg(J(\mathfrak{u})) + 2$, where card is the set cardinality and deg indicates the polynomial degree. The evaluation of u_t^{nl} through (8) - to be done online - is thus extremely fast, since it just requires to find the real roots of a polynomial whose analytical expression is known and to compute the objective function for a "small" number of values.

The linear controller K^{lin} is defined by the extended PID (Proportional Integral Derivative) control law

$$u_t^{lin}(\theta) = u_{t-1}^{lin}(\theta) + \sum_{i=0}^{n_\theta} \theta_i e_{t-i} \tag{9}$$

where $e_t = r_t - y_t$ is the tracking error, n_θ is the controller order and the θ_i's denote the controller parameters. Note that, for $n_\theta = 1$ and $n_\theta = 2$, the standard PI and PID controller are selected, respectively. The linear controller design is based on the Virtual Reference Feedback Tuning (VRFT) method, [1], slightly adapted for the present setting as indicated in [20].

III. Closed-loop stability analysis

The resulting closed-loop feedback system of Figure 1 turns out to be described by

$$\begin{aligned} y_{t+1} &= g(\boldsymbol{y}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_t) \\ u_t &= u_t^{nl} + u_t^{lin} \\ u_t^{nl} &= K^{nl}\left(r_{t+1}, \boldsymbol{y}_t, \boldsymbol{u}_{t-1}^{nl}\right) \\ u_t^{lin} &= K^{lin}\left(\boldsymbol{r}_t - \boldsymbol{y}_t, \boldsymbol{u}_{t-1}^{lin}\right) \end{aligned} \tag{10}$$

where K^{nl} and K^{lin} are defined in (8) and (9), respectively, and $u_t \in U$, $\forall t$. Let us choose the initial condition $\boldsymbol{r}_0 = \boldsymbol{y}_0$.

In this section, we study the stability properties of such a system, according to the following stability notion.

Definition 1: A nonlinear system (possibly time-varying), with inputs r_t and ξ_t, and output y_t, is *finite-gain ℓ_∞ stable* on $\left(\mathcal{Y}^0, \mathcal{R}, \Xi\right)$ if finite and non-negative constants Γ_r, Γ_ξ and Λ exist such that

$$\|\boldsymbol{y}\|_\infty \le \Gamma_r \|\boldsymbol{r}\|_\infty + \Gamma_\xi \|\boldsymbol{\xi}\|_\infty + \Lambda$$

for any $(\boldsymbol{y}_0, \boldsymbol{r}, \boldsymbol{\xi}) \in \mathcal{Y}^0 \times \mathcal{R} \times \Xi$. □

Note that this finite-gain stability definition is more general than the standard one, which is obtained for $\mathcal{R} = \ell_\infty$ and $\Xi = \ell_\infty$, see e.g. [11].

Before studying how to guarantee finite-gain stability of the feedback system (10), some additional assumptions are introduced and discussed.

Assumption 1 (Lipschitzianity): The function g in (1) and (10) is *Lipschitz continuous* on $Y^n \times U^n \times \Xi^n$. Without loss of generality, it is also assumed that $Y^n \times U^n \times \Xi^n$ contains the origin. □

This assumption is mild, since most real-world dynamic systems are described by functions that are Lipschitz continuous on a compact set.

From Assumption 1, it follows that g can be written as

$$g(\boldsymbol{y}_t, \boldsymbol{u}_t, \boldsymbol{\xi}_t) = g^o(\boldsymbol{y}_t, \boldsymbol{u}_t) + g_t^\xi \boldsymbol{\xi}_t$$

where $g^o(\boldsymbol{y}_t, \boldsymbol{u}_t) \doteq g(\boldsymbol{y}_t, \boldsymbol{u}_t, \boldsymbol{0})$ and $g_t^\xi \in \mathbb{R}^{1 \times n}$ is a time-varying parameter (dependent on $\boldsymbol{y}_t$, $\boldsymbol{u}_t$ and $\boldsymbol{\xi}_t$) bounded on $Y^n \times U^n \times \Xi^n$ as $\left\| g_t^\xi \right\|_\infty \le \gamma_\xi$, for some $\gamma_\xi < \infty$. Assumption 1, together with (4), implies that the residue function

$$\Delta(\boldsymbol{y}_t, \boldsymbol{u}_t) \doteq g^o(\boldsymbol{y}_t, \boldsymbol{u}_t) - f(\boldsymbol{y}_t, \boldsymbol{u}_t)$$

is Lipschitz continuous on $Y^n \times U^n$. Indeed, Δ is the sum of two functions that are Lipschitz on the compact domain $Y^n \times U^n$; g^o is Lipschitz by assumption, while f is Lipschitz by construction, since any (multivariate) polynomial is Lipschitz on any compact set. Hence, a finite and non-negative constant γ_y exists, such that

$$|\Delta(y, u) - \Delta(y', u)| \le \gamma_y \|y - y'\|_\infty$$

for all $y, y' \in Y^n$ and all $u \in U^n$.

Assumption 2 (Model accuracy): $\gamma_y < 1$. □

The meaning of this assumption is clear: it requires f to accurately describe the variability of g with respect to $\boldsymbol{y}_t$. Now, consider that

$$\begin{aligned} &\hat{e}_{t+1} \doteq r_{t+1} - \hat{y}_{t+1} = r_{t+1} - f(\boldsymbol{y}_t, \boldsymbol{u}_t) \\ &\equiv r_{t+1} - f\left(\boldsymbol{y}_t, \boldsymbol{u}_t\left(r_{t+1}, \boldsymbol{r}_t, \boldsymbol{y}_t, \boldsymbol{u}_{t-1}^{nl}, \boldsymbol{u}_{t-1}^{lin}\right)\right) \\ &\doteq F(\boldsymbol{y}_t, \boldsymbol{r}_t, \boldsymbol{v}_t), \end{aligned}$$

where $\boldsymbol{v}_t \doteq (r_{t+1}, \boldsymbol{u}_{t-1}^{nl}, \boldsymbol{u}_{t-1}^{lin}) \in V$ and V is a compact set. Then, for any $(\boldsymbol{y}_t, \boldsymbol{r}_t, \boldsymbol{v}_t) \in Y^n \times R^n \times V^n$,

$$|\hat{e}_{t+1}| \le \Gamma_y \|\boldsymbol{y}_t\|_\infty + \Gamma_s \|\boldsymbol{r}_t\|_\infty + \Lambda_e \tag{11}$$

where $\Gamma_y, \Gamma_s, \Lambda_e < \infty$. This inequality directly follows from the fact that the model function f is Lipschitz continuous on $Y^n \times U^n$. Note that (11) does not imply that $\boldsymbol{y} \in Y^\infty$. This inequality just ensures that, if $(\boldsymbol{y}_t, \boldsymbol{r}_t, \boldsymbol{v}_t) \in Y^n \times R^n \times V^n$ for all t, then the signal $\hat{\boldsymbol{e}}$ is bounded as in (11) , with given constants $\Gamma_y, \Gamma_s, \Lambda_e < \infty$.

Assumption 3 (Effective model inversion): $\Gamma_y \le 1 - \gamma_y$. □

This assumption is not restrictive: it is certainly satisfied if $\mu = 0$ and the reference $\boldsymbol{r} = (r_1, r_2, \ldots)$ is a model solution (i.e. r_{t+1} is in the range of $f(\boldsymbol{y}_t, \cdot)$ for all t). Indeed, in this case, $\hat{y}_{t+1} = r_{t+1}$, $\forall t$, since K^{nl} performs an exact inversion of the model, see again (6) (K^{lin} gives a null input signal in this case). This implies that $\Gamma_y = 0$, $\Gamma_s = 0$ and $\Lambda_e = 0$. Hence, if a sufficiently small μ is chosen and the reference is sufficiently close to a system solution, supposing that inequality (11) holds with a sufficiently small Γ_y is reasonable. Practical indications for choosing μ are given at the end of Section IV.

The meaning of Assumption 3 is that, in order to guarantee closed-loop stability, the controller must perform an effective right-inversion of the model and this inversion should depend as less as possible on the current working point $\boldsymbol{y}_t$. In particular, the bound (11) implies that, if the model (3) is exact and the plant is characterized by a sufficient control authority, the designed controller stabilizes the closed-loop system (a direct consequence of Theorem 1 below). In cases where the the control authority is not sufficient, it may not be possible to accurately invert the model. This may result in a value of Γ_y larger than $1-\gamma_y$, thus precluding the possibility of stabilizing the closed-loop system. Note that the command authority is reduced if large values of μ are required to satisfy physical constraints that may be present in the plant and/or in the actuators.

To conclude this discussion about Assumption 3, we remark that the issues related to control authority are common to all control methods: if the authority is severely limited for some reason, then it may not be possible to stabilize the plant, whatever is the control method used.

Now, to formulate our last assumption, define

$$\bar{e} \doteq \frac{1}{1-\lambda_y}\left(\lambda_r \bar{r} + \gamma_\xi \bar{\xi} + \Lambda_g\right) \tag{12}$$

where $\lambda_y \doteq \Gamma_y + \gamma_y < 1$, $\lambda_r \doteq \lambda_y + \Gamma_s$ and $\Lambda_g \doteq \Lambda_e + \max_{u\in U^n} |\Delta(\mathbf{0}, u)|$. Note that $\bar{e}$ is bounded, being the sum of bounded quantities. In particular, for null disturbances ($\bar{\xi}=0$), exact modeling ($f=g$, $\Delta=0$, $\gamma_y=0$) and reference signals properly chosen ($\Gamma_y=0$, $\Gamma_s=0$, $\Lambda_e=0$), we have $\bar{e}=0$. In realistic situations, with reasonable disturbances, sufficiently accurate modeling and reference signals properly chosen, $\bar{e}$ can be reasonably "small" (that is, $\bar{e} \ll \bar{r}$).

Assumption 4 (Output domain exploration):
$\bar{y} \geq \bar{r} + \bar{e}$. □

This assumption requires that the set Y explored by the output data is somewhat larger than the set R where the trajectory of interest are defined. Note that it can always be met just collecting data that sufficiently enlarge the set Y.

Closed-loop stability of the system (10) is stated by the following result, which also provides a reliable bound on the tracking error.

Theorem 1: Consider the system (10) and let Assumptions 1-4 hold. Then:
(i) The feedback system (10), having inputs r_t and ξ_t and output y_t, is finite-gain ℓ_∞ stable on $\left(\mathcal{Y}^0, \mathcal{R}, \Xi\right)$.
(ii) The tracking error signal $\boldsymbol{e} \doteq \boldsymbol{r} - \boldsymbol{y}$ is bounded as

$$\|\boldsymbol{e}\|_\infty \leq \bar{e}. \tag{13}$$

Proof. The proof of the theorem is structured as follows. Firstly, the tracking error is proven to be upper bounded by a suitable combination of the norms of the output and the reference. Secondly, it is shown that, under the assumption of an effective model inversion, such a bound is equivalent to a bound on the tracking error, whatever the output is (claim (ii)). Claim (i) is derived as a straightforward consequence of claim (ii).

To start with, consider that

$$e_{t+1} \doteq r_{t+1} - y_{t+1} = \hat{e}_{t+1} - \delta y_t$$

where

$$\begin{aligned} \hat{e}_{t+1} &= r_{t+1} - \hat{y}_{t+1} = F\left(\boldsymbol{y}_t, \boldsymbol{r}_t, \boldsymbol{v}_t\right) \\ \delta y_t &= \Delta\left(\boldsymbol{y}_t, \boldsymbol{u}_t\right) + g_t^\xi \boldsymbol{\xi}_t. \end{aligned}$$

The term $\hat{e}_{t+1}$ is bounded according to (11). Note that $\|\boldsymbol{y}_t\|_\infty$ could be unbounded. In order to derive a bound on δy_t, we can use Assumption 2 and observe that, for any $\boldsymbol{y}_t \in Y^n$,

$$\begin{aligned} &|\Delta\left(\boldsymbol{y}_t, \boldsymbol{u}_t\right)| - |\Delta\left(\mathbf{0}, \boldsymbol{u}_t\right)| \\ &\leq |\Delta\left(\boldsymbol{y}_t, \boldsymbol{u}_t\right) - \Delta\left(\mathbf{0}, \boldsymbol{u}_t\right)| \leq \gamma_y \|\boldsymbol{y}_t\|_\infty. \end{aligned}$$

The following inequality thus holds for any $\boldsymbol{y}_t \in Y^n$:

$$\begin{aligned} |\delta y_t| &\leq |\Delta\left(\boldsymbol{y}_t, \boldsymbol{u}_t\right)| + \gamma_\xi \|\boldsymbol{\xi}_t\|_\infty \\ &\leq \gamma_y \|\boldsymbol{y}_t\|_\infty + \gamma_\xi \|\boldsymbol{\xi}_t\|_\infty + \bar{\Delta}, \end{aligned}$$

where $\bar{\Delta} \doteq \max_{u\in U^n} |\Delta(\mathbf{0}, u)| < \infty$. Hence,

$$\begin{aligned} |e_{t+1}| \leq \lambda_y \|\boldsymbol{y}_t\|_\infty + \Gamma_s \|\boldsymbol{r}_t\|_\infty + \Lambda_e \\ + \gamma_\xi \|\boldsymbol{\xi}_t\|_\infty + \bar{\Delta}, \end{aligned} \tag{14}$$

which proves that the tracking error e is bounded by a suitable combination of the norms of $\boldsymbol{y}_t$ and $\boldsymbol{r}_t$, for any $\boldsymbol{y}_t \in Y^n$. Note that (14) in this form is of no use, since $\|\boldsymbol{y}_t\|_\infty$ could be unbounded and thus the condition $\boldsymbol{y}_t \in Y^n$ may not hold. However, (14) can be rewritten as

$$\begin{aligned} |e_{t+1}| \leq \lambda_y \|\boldsymbol{e}_t\|_\infty + \lambda_y \|\boldsymbol{r}_t\|_\infty + \Gamma_s \|\boldsymbol{r}_t\|_\infty \\ + \gamma_\xi \|\boldsymbol{\xi}_t\|_\infty + \Lambda_g, \end{aligned} \tag{15}$$

that means,

$$|e_{t+1}| \leq \lambda_y \|\boldsymbol{e}_t\|_\infty + w, \tag{16}$$

where $w \doteq \lambda_r \bar{r} + \gamma_\xi \bar{\xi} + \Lambda_g$. Inequality (16), again, holds only if $\boldsymbol{y}_t \in Y^n$.

Consider now that, by assumption, $\boldsymbol{y}_0 \in R^n \subseteq Y^n$. This implies that, at time $t=0$, inequality (16) holds. Being $\boldsymbol{e}_0 = \boldsymbol{r}_0 - \boldsymbol{y}_0 = 0$ for the selected initialization of $\boldsymbol{r}_0$, we have

$$|e_1| \leq \lambda_y \|\boldsymbol{e}_0\|_\infty + w = w \leq \bar{e}.$$

Since $|e_1| \leq \bar{e}$ and $|r_1| \leq \bar{r}$, it follows from Assumption 4 that $y_1 \in Y$. Consequently, the Lipschitzianity assumption holds and (16) can be used also for $t=1$, giving

$$\begin{aligned} |e_2| &\leq \lambda_y |e_1| + w \leq \lambda_y w + w \\ &\leq w \textstyle\sum_{k=0}^{\infty} \lambda_y^k \leq \frac{w}{1-\lambda_y} = \bar{e} \end{aligned}$$

where the geometric series sum has been obtained thanks to the fact that, by Assumption 3, $\lambda_y < 1$. It follows that $y_2 \in Y$ and, consequently, (16) can be used also for $t=2$. Iterating the above reasoning,

$$\begin{aligned} |e_3| &\leq \lambda_y \max\{|e_2|, |e_1|\} + w \\ &\leq \lambda_y |e_2| + w \leq \lambda_y^2 w + \lambda_y w + w \leq \bar{e} \\ &\vdots \\ |e_{t+1}| &\leq w \textstyle\sum_{k=0}^{t} \lambda_y^k \\ &\leq w \textstyle\sum_{k=0}^{\infty} \lambda_y^k \leq \frac{w}{1-\lambda_y} = \bar{e}. \end{aligned}$$

Then, $y_t \in Y$, $\forall t \geq 0$ and (13) holds (claim (ii)). Claim (i) is a direct consequence of claim (ii) and the relation $\boldsymbol{y} = \boldsymbol{r} - \boldsymbol{e}$. □

Theorem 1 can be interpreted as follows. Two main conditions are sufficient to guarantee closed-loop stability. First, the model must describe accurately the model rate of variation

with respect to $\boldsymbol{y}_t$ (Assumption 2). Second, the controller has to perform an effective inversion of the model (Assumption 3). These conditions allow for closed-loop stability and lead to the tracking error bound given in Theorem 1. It can be noted that this error is reduced if the model provides a "small" prediction error (Δ measures of the prediction error). Hence, the model should satisfy two requirements: it must be accurate (on a sufficiently large set) in describing the dependence on $\boldsymbol{y}_t$ and, at the same time, in reproducing the system output. Note that, in the proposed control scheme, the model does not work in simulation but *in prediction*.

Besides choosing $r_t \in R$, $\forall t$, an indication for generating suitable references can be the following: a reference signal should be a solution (or an approximate solution) of the system to control, i.e. a signal $\boldsymbol{r} = (r_1, r_2, \ldots)$ for which, at each time t, a u_t exists giving $y_{t+1} = g\left(\boldsymbol{y}_t, u_t, u_{t-1}, \ldots, u_{t-n+1}, \boldsymbol{\xi}_t\right) \cong r_{t+1}$. As discussed above, this indication allows one to have a "small" $\bar{e}$, thus enlarging the set of feasible references and reducing the tracking error. More in general, the reference trajectory must be compatible with the physical properties of the system to control. For instance, in a second order mechanical system, the output is typically a position. Thus, the reference can be generated as a sequence of values ranging in the physical domain of this variable with reasonable variations (no other particular indications are required). Note anyway that reference design is a well-known open problem, arising for most nonlinear control methods.

Remark 1: The constants γ_y, Γ_y and $\bar{e}$ can be estimated from the available data by means of the validation method in [16], allowing us to verify in practice the stability conditions of Theorem 1. We will see in Section IV that the most critical of these conditions (i.e., $\gamma_y < 1$ and $\Gamma_y \leq 1 - \gamma_y$) can also be *imposed a-priori* by our model identification algorithm. □

Remark 2: The present stability analysis accounts for the fact that the command input is constrained in the set U. Indeed, it may be guessed that, due to the command constraints, the control action may be too small to stabilize the system. Theorem 1 addresses this problem: it implies that, if the initial condition is in the region where the residue function is Lipschitz (with a proper constant) and the reference is suitably chosen, then the situation where the control action is not large enough never occurs. Clearly, as discussed below Assumption 3, the plant to control must be characterized by a sufficient control authority. □

IV. Control-oriented model identification

In this section, an algorithm is proposed for identifying the prediction model (3) needed by the control law (6). The algorithm is an evolution of the one in [20], properly modified to incorporate a constraint enforcing closed-loop stability.

To start with, choose a set of polynomial basis functions $\{\phi_i, i = 1, \ldots, N\}$. Define

$$\tilde{\boldsymbol{y}} \doteq \left(\tilde{y}_{t_1+1}, \ldots, \tilde{y}_{t_2+1}\right)$$

$$\Phi \doteq \begin{bmatrix} \phi_1\left(\tilde{\boldsymbol{y}}_{t_1}, \tilde{\boldsymbol{u}}_{t_1}\right) & \cdots & \phi_N\left(\tilde{\boldsymbol{y}}_{t_1}, \tilde{\boldsymbol{u}}_{t_1}\right) \\ \vdots & \ddots & \vdots \\ \phi_1\left(\tilde{\boldsymbol{y}}_{t_2}, \tilde{\boldsymbol{u}}_{t_2}\right) & \cdots & \phi_N\left(\tilde{\boldsymbol{y}}_{t_2}, \tilde{\boldsymbol{u}}_{t_2}\right) \end{bmatrix}$$

where $t_1 \doteq 1 - L + n$, $t_2 \doteq -1$, and $\tilde{u}_t$ and $\tilde{y}_t$ are the input-output measurements of the data set (2). Consider the set $SC \subset \mathbb{R}^N$, defined as

$$\begin{aligned} SC(\gamma, \eta, \rho) \doteq \{\beta : &|\widetilde{y}_{l+1} - \widetilde{y}_{k+1} + (\boldsymbol{\Phi}_k - \boldsymbol{\Phi}_l)\,\beta| \\ &< \gamma\rho \left\|\widetilde{\boldsymbol{y}}_l - \widetilde{\boldsymbol{y}}_k\right\|_\infty + 2\eta\rho, k \in \mathcal{T}, l \in \Upsilon_k\} \end{aligned}$$

where $\mathcal{T} \doteq \{t_1, \ldots, t_2\}$, $\boldsymbol{\Phi}_k$ indicates the kth row of Φ, Υ_k is the set of indexes given by

$$\Upsilon_k \doteq \{i : \left\|\tilde{\boldsymbol{u}}_k - \tilde{\boldsymbol{u}}_i\right\|_\infty \leq \zeta\}$$

and ζ is the minimum value for which every set Υ_k contains at least two elements. The set SC is defined by a set of linear inequalities in β and is thus convex in β. This set has been introduced in [19] and is used in order to enforce closed-loop stability.

The parameter vector $\alpha \doteq (\alpha_1, \ldots, \alpha_N)$ of the model defined by (3) and (4) can be identified by means of the following algorithm, completely based on convex optimization. Note that most of the required parameters are chosen by the algorithm itself, without requiring extensive trial and error procedures.

Algorithm

1) Initialization: choose "low" model order (e.g., $n = 1$) and polynomial degree (e.g., 2); choose a precision level η_0 (e.g., $\eta_0 = 0.05 \left\|\tilde{\boldsymbol{y}}\right\|_\infty$).
2) Construct the vector $\tilde{\boldsymbol{y}}$ and the matrix Φ as indicated above.
3) Let $\eta \doteq \max\left(\eta_0, \eta_1\right)$, where
$$\eta_1 = \min_{\beta \in \mathbb{R}^N} \left\|\tilde{\boldsymbol{y}} - \boldsymbol{\Phi}\beta\right\|_\infty .$$
4) Consider the optimization problem
$$\begin{aligned} &\alpha = \arg \min_{\beta \in \mathbb{R}^N} \left\|\beta\right\|_1 \\ &\text{subject to} \\ &(a)\ \beta \in SC(\gamma_y, \eta, \rho) \\ &(b)\ \left\|\tilde{\boldsymbol{y}} - \boldsymbol{\Phi}\beta\right\|_\infty \leq \eta\rho \end{aligned} \tag{17}$$
where $\gamma_y < 1$ and ρ is a real number slightly larger than 1 (e.g., $\rho = 1 + \delta\rho$, $\delta\rho = 0.05$).
If the optimization problem (17) is feasible, solve it, return α and stop. Else, increase the model order n and go to step 2. The order should be increased up to a maximum value n_{max}, chosen on the basis of some rough knowledge on the order of system to control (this kind of knowledge is available in most practical cases).
5) If $n = n_{max}$ and (17) is not feasible, repeat steps 2-4 for increasing polynomial degree d. The degree should be increased up to a maximum value d_{max}. Note that this choice is not critical thanks to ℓ_1 norm minimization in (17), which penalizes models with a large number of basis functions.
6) If $n = n_{max}$, $d = d_{max}$ and (17) is not feasible, repeat steps 1-5 for $\rho = 1 + 2\delta\rho, 1 + 3\delta\rho, \ldots$.

The algorithm allows the achievement of three important features:

i. ***Closed-loop stability***. As proven in [19], under reasonable conditions, constraint (a) ensures that the function $\Delta \doteq g - f$ has a Lipschitz constant smaller than γ_y for a sufficiently large L. On the other hand, Theorem 1 shows that having this constant smaller than 1 is a key condition for closed-loop stability. The theorem also requires that $\Gamma_y \leq 1 - \gamma_y$, where Γ_y is a constant measuring the inversion capability of the controller (see again (11) and the subsequent discussion). Since the value of Γ_y can be reasonably estimated and in a certain way tuned (see the discussion below), it can be concluded that Algorithm 1 can ensure closed-loop stability when the number of data is sufficiently large.

ii. ***"Small" tracking error***. Constraint (b) is aimed at providing a model with a "small" prediction error (this error, evaluated on the design data set, is given by $\|\widetilde{\mathbf{y}} - \mathbf{\Phi}\alpha\|_\infty$). As shown in Theorem 1, reducing this error allows us to obtain a "small" tracking error. Note that there is a trade-off between stability and tracking performance: In step 6, ρ is increased until the stability condition is met. However, increasing ρ causes an increase of the prediction error and, consequently, of the tracking error.

iii. ***Model sparsity***. In (17), the ℓ_1 norm of the coefficient vector β is minimized, leading to a sparse coefficient vector α, i.e. a vector with a "small" number of non-zero elements, [26], [17]. Sparsity is important to ensure a low complexity model, limiting at the same time well known issues such as over-fitting and the curse of dimensionality. Sparsity allows also an efficient implementation of the model/controller on real-time processors, which may have limited memory and computation capacities.

Once a model has been identified, the nonlinear controller K^{nl} is obtained by its inversion, as explained in Section II. Only one design parameter needs to be chosen for this inversion: the weight μ in (7). If no particular requirements on the activity of the command input u_t have to be satisfied, the simplest choice is $\mu = 0$. Otherwise, if the input activity has to be reduced, a value $0 < \mu \leq \bar{\mu}$ can be chosen, where $\bar{\mu}$ is the maximum value for which the stability condition $\Gamma_y \leq 1 - \gamma_y$ holds. This condition can be checked (approximately) by deriving an estimate $\hat{\Gamma}_y$ of Γ_y from the data (2) as follows. Let

$$\mathcal{D}^\Gamma \doteq \{\tilde{w}_t, \hat{y}_{t+1}\}_{t=1-L+m}^{-1} \tag{18}$$

where

$$\begin{aligned} \hat{y}_t &= f\left(\tilde{\boldsymbol{y}}_{t-1}, \tilde{\boldsymbol{u}}_{t-1}\right) \\ \tilde{u}_{t-1} &= K^{nl}\left(\tilde{y}_t, \tilde{\boldsymbol{q}}_{t-1}\right) + K^{lin}\left(\tilde{y}_t, \tilde{\boldsymbol{q}}_{t-1}\right) \\ \tilde{\boldsymbol{q}}_{t-1} &= \left(\tilde{y}_{t-1}, \ldots, \tilde{y}_{t-n}, \tilde{u}_{t-2}^{nl}, \ldots, \tilde{u}_{t-n}^{nl}\right) \\ \tilde{w}_t &= \left(\tilde{y}_t, \ldots, \tilde{y}_{t-m+1}\right), \end{aligned} \tag{19}$$

$\tilde{u}_t$ and $\tilde{y}_t$ are the input-output measurements of the data set (2), and $m \gg n$. The estimate $\hat{\Gamma}_y$ can be obtained applying the validation method of [16] to the data set (18). Observing that $\tilde{u}_{t-1}^{nl} \equiv \tilde{u}_{t-1}^{nl}(\mu)$ and thus $\hat{\Gamma}_y \equiv \hat{\Gamma}_y(\mu)$, μ must be chosen in such a way that

$$\hat{\Gamma}_y(\mu) \leq 1 - \gamma_y. \tag{20}$$

Note that, if correctly designed, the linear controller enhances the inversion properties. Therefore, if condition (20) is satisfied when only the nonlinear controller appears in (19), we expect this condition to be satisfied also adding a suitable linear controller to (19). If this does not happen, then the linear controller is not proper and must be re-designed.

Remark 3: The stability conditions $\gamma_y < 1$ and $\Gamma_y \leq 1 - \gamma_y$ of Theorem 1 can give indications on the choice of the control system sampling time T_s. As discussed in [7], a too small T_s leads to models where $\hat{y}_{t+1} \cong y_t$. These kinds of models have a strong dependence on past outputs and a weak dependence on the input, resulting in large values of γ_y and Γ_y. It is thus expected that γ_y and Γ_y can be reduced by increasing T_s. Clearly, to capture the relevant dynamics of the system and allow a prompt control action, T_s must be not too large, in the sense of the Nyquist-Shannon sampling theorem. □

V. Example: Control of the Duffing Oscillator

The simulation example presented in this section is about control of the Duffing system, a second-order damped oscillator with nonlinear spring. The goals of this example are to test the control design method presented in this paper and to compare it with the one in [20] (i.e., with the D^2-IBC approach without the additional constraint in the identification algorithm to enforce closed-loop stability).

The Duffing system is a nonlinear oscillator defined by the following equations:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\alpha_1 x_1 - \alpha_2 x_1^3 - \beta x_2 + u \\ y &= x_1 + \xi \end{aligned} \tag{21}$$

where $x = (x_1, x_2)$ is the state, u is the input, y is the output, and ξ is a noise. The state variables x_1 and x_2 represent the oscillator position and velocity, respectively. The following values of the parameters have been assumed: $\alpha_1 = -1$, $\alpha_2 = 1$, $\beta = 0.2$. For these parameter values and for certain input signals, this system exhibits a chaotic behavior, and this makes its control a particularly challenging problem.

A. D^2-IBC design and test

A simulation of the Duffing system (21) having duration 200 s was performed, using the input signal $u(\tau) = 0.3\sin(\tau) + \xi^u(\tau)$, where τ here denotes the continuous time and ξ^u is a white Gaussian noise with zero mean and standard deviation 0.2. Notice that the selected input signal is persistently exciting in the sense of [12]. This is mandatory, as the design of the linear controller in D^2-IBC has been reformulated as a pure linear system identification problem, and therefore linear identifiability conditions must be satisfied.

A set of $L = 2000$ data was collected from this simulation, with a sampling time $T_s = 0.1$ s: $\mathcal{D} \doteq \{\tilde{u}_t, \tilde{y}_t\}_{t=-L+1}^{0}$ where $\tilde{u}_t = u(T_s t)$ are the measurements of the input and $\tilde{y}_t = y(T_s t)$ are the measurements of the output. ξ was simulated as a zero-mean Gaussian noise with a noise-to-signal standard deviation ratio (NSR) of 0.03.

A nonlinear controller K^{nl} was designed from the collected data, following the approach of Sections II-IV. The value $\mu = 0.01$ was chosen in the cost function (7). The basis

functions were generated as products of univariate polynomials with maximum degree 2, yielding a set of 28 functions with maximum degree 4. Then, Algorithm 1 was run, choosing $\eta_0 = 0.001$, $\gamma_y = 0.8$, $\rho = 1.05$, $n_{max} = 6$ and $d_{max} = 8$. The following parameter values were produced by the algorithm: $n = 2$, $d = 4$ and $\eta = 0.18$. The constant Γ_y was estimated as discussed in Section IV and the value 0.07 was obtained, showing that the stability conditions given by Assumptions 2 and 3 are satisfied. The algorithm selected 23 of the initial 28 basis functions. We can observe that the sparsification properties of the algorithm can be more important in situations where the regressor is of larger dimensions and the polynomial degree is higher, involving hundreds or thousands basis functions.

A PID linear controller K^{lin} was then designed following the VRFT approach, [1] as indicated in [20].

The control scheme of Figure 1 was implemented, where S is the system (21), K^{nl} and K^{lin} are the designed controllers, and ξ_t is a noise with a NSR of 0.03.

A simulation test of the control system with duration 800 s was performed, using zero initial conditions and a reference signal r_t generated as a sequence of random steps, filtered by a second-order filter with a cutoff frequency of 2 rad/s (this filter was inserted in order to avoid abrupt variations). In Figure 2, the output of the D^2-IBC control system is compared to the reference. The command input signal u is also shown in this figure. To quantify the performance level and the command activity of the controller, the Root Mean Square (RMS) tracking error and RMS input were considered, defined as

$$\begin{aligned} RMS_e &\doteq \sqrt{\tfrac{1}{8000}\textstyle\sum_{t=1}^{8000}(r_t - y_t)^2} \\ RMS_u &\doteq \sqrt{\tfrac{1}{8000}\textstyle\sum_{t=1}^{8000}u_t^2}. \end{aligned} \tag{22}$$

The values obtained in this first test are $RMS_e = 0.0161$ and $RMS_u = 0.252$. From these results, it can be observed that the tracking accuracy is satisfactory and the command activity is quite limited. The command signal is subject to some chattering but this is normal since a non-negligible noise is affecting the output used for feedback.

B. Monte Carlo simulations

A Monte Carlo (MC) simulation was carried out, where the above data-generation-control-design-and-testing procedure was repeated 100 times. For each trial, the controller performance was evaluated by means of the RMS indexes (22). This MC simulation was repeated three times, one for each of the following values of NSR: 0.03, 0.06, 0.1. In each trial, the same NSR has been adopted for both the data generation phase and the control test phase.

For comparison, a similar MC simulation was performed using other D^2-IBC controllers, designed by means of the algorithm in [20]. These controllers are similar to the ones described here but obtained without imposing the stability constraint (a) in (17). For completeness, other comparisons can be found in [20], e.g. with DFK [19] and VRFT.

The average values of RMS_e and RMS_u obtained by the designed controllers in the MC simulations are reported

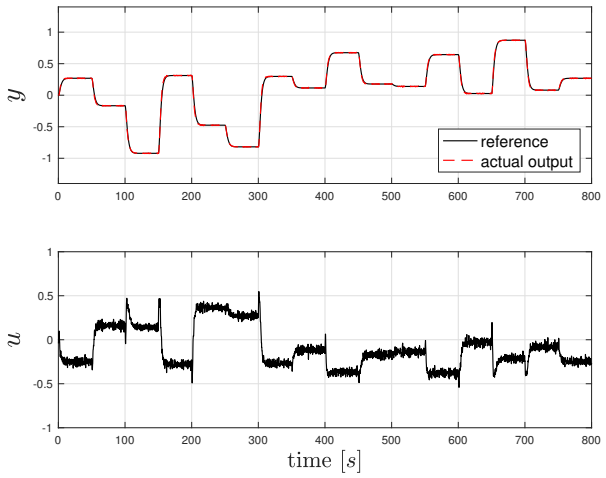

Fig. 2. Tracking performance of a D^2-IBC$_c$ control system.

	D^2-IBC$_c$		D^2-IBC$_{uc}$	
NSR	RMS_e	RMS_u	RMS_e	RMS_u
0.03	0.0169	0.253	0.0164	0.255
0.06	0.0312	0.273	0.0301	0.276
0.1	0.0442	0.294	0.2963	0.315

TABLE I
AVERAGE RMS FOR DIFFERENT NOISE LEVELS.

in Table I. D^2-IBC$_c$ indicates the controllers obtained imposing the stability constraint, whereas D^2-IBC$_{uc}$ indicates the controllers obtained without the stability constraint. From these results, it can be concluded that, on the one hand, all the nonlinear controllers provide a satisfactory performance when the noise magnitude is below $6\% - 7\%$, notwithstanding the additional constraint represents only a sufficient (then possibly conservative) condition. On the other hand, for large noise levels, the D^2-IBC$_c$ controllers still show a satisfactory performance, while the D^2-IBC$_{uc}$ controllers lead to larger tracking errors. An unstable behavior was also sometimes observed for the D^2-IBC$_{uc}$ controllers. This confirms the theoretical results of Section III, which show that imposing the constraint (a) in (17) enforces the stability of the closed-loop system.

C. Sensitivity analysis

To test the sensitivity of the D^2-IBC design with respect to the choice of the command activity weight μ, values $\mu \in \{10^{-3}, 10^{-1}, 1, 5\}$ were considered. For each value of μ, a MC simulation was carried out, where the data-generation-control-design-and-testing procedure of Section V-A was repeated 100 times. The values of all the parameters, except μ, were maintained equal to those of Section V-A.

As shown in Table II, the results obtained with $\mu \in \{10^{-3}, 10^{-1}, 1\}$ are very similar to those obtained with $\mu = 0.01$ in terms of RMS_e. With $\mu = 0.001$, the command signal is a little bit more nervous, while with larger values of μ it is smoother. However, the values of RMS_u do not change substantially in function of μ. The reason is that, for a large μ, the linear controller is characterized by a higher activity,

	D^2-IBC$_c$	
μ	RMS_e	RMS_u
10^{-3}	0.0159	0.276
10^{-2}	0.0169	0.253
10^{-1}	0.0167	0.251
1	0.0171	0.246
5	0.0362	0.249

TABLE II
AVERAGE RMS FOR DIFFERENT VALUES OF μ.

	D^2-IBC$_c$	
T_s	RMS_e	RMS_u
0.01	0.0112	0.248
0.05	0.0135	0.251
0.1	0.0169	0.253

TABLE III
AVERAGE RMS FOR DIFFERENT VALUES OF T_s.

in order to compensate the reduced activity of the nonlinear controller. With $\mu = 5$, a slightly diverging behavior was observed sometimes. This kind of behavior is consistent with Theorem 1. It interesting to note that, with $\mu = 1$, Assumption 3 is not satisfied but the closed-loop system showed a stable behavior in all the simulations. This is not surprising, as the assumption gives only a sufficient condition.

To test the sensitivity of the D^2-IBC design with respect to the sampling time choice, values $T_s \in \{0.01, 0.05\}$ s were considered. For each value of T_s, a MC simulation was carried out, where the data-generation-control-design-and-testing procedure of Section V-A was repeated 100 times. The values of all the parameters, except T_s, were maintained equal to those of Section V-A.

As shown in Table III, the results obtained with $T_s \in \{0.01, 0.05\}$ s are very similar to those obtained with $T_s = 0.1$ s in terms of both RMS_e and RMS_u.

In conclusion, in all the MC simulations carried out with different values of μ and T_s, the D^2-IBC controllers showed satisfactory robustness properties. Another feature of the D^2-IBC approach that was evidenced by this robustness analysis is that the activities of the nonlinear and linear controllers are coordinated, in the sense that, when one of these controllers reduces its activity, the other is able to compensate this reduction, in order to maintain the desired performance level.

VI. CONCLUSION

In this paper, we address the stability problem in the D^2-IBC approach. The controller returned by the original algorithm of [20] is in fact not theoretically guaranteed to stabilize the closed-loop system. More specifically, we provide a sufficient finite-gain stability condition and we prove that it can be verified if a suitable constraint is added to the model optimization task. In the modified version of the D^2-IBC method proposed here, the model is then identified so that the controller computed out of that model is guaranteed to stabilize the system. Practical applications of the D^2-IBC approach include braking control [4], cancer immunotherapy control [21], control of air and charging systems of diesel engines [15], control of a 2-DOF robot manipulator [18], glucose control in type 1 diabetic patients [18].

REFERENCES

[1] 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.

[2] M.C. Campi and S.M. Savaresi. Direct nonlinear control design: The virtual reference feedback tuning (vrft) approach. *IEEE Transactions on Automatic Control*, 51(1):14–27, 2006.

[3] S. Formentin, C. Novara, S.M. Savaresi, and M. Milanese. Active braking control system design: The D2-IBC approach. *IEEE/ASME Transactions on Mechatronics*, 20(4):1573–1584, 2015.

[4] S. Formentin, C. Novara, S.M. Savaresi, and M. Milanese. Active braking control system design: the D2-IBC approach. *IEEE/ASME Transactions on Mechatronics*, 20(4):1573–1584, 2015.

[5] A. Freeman and V. Kokotovic. *Robust Nonlinear Control Design*. Birkhuser, Boston, 1996.

[6] M. Gevers. Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, 11(4):335–352, 2005.

[7] G.C. Goodwin, J.I. Yuz, J.C. Aguero, and M. Cea. Sampling and sampled-data models. In *American Control Conference, plenary lecture*, Baltimore, MD, USA, 2010.

[8] G. Guardabassi and S.M. Savaresi. Approximate linearization via feedback—an overview. *Automatica*, 37(1):1–15, 2001.

[9] V. Hagenmeyer and E. Delaleau. Exact feedforward linearization based on differential flatness. *International Journal of Control*, 76(6):537–556, 2003.

[10] H. Hjalmarsson. From experiment design to closed-loop control. *Automatica*, 41(3):393–438, 2005.

[11] H.K. Khalil. *Nonlinear Systems*. Prentice Hall, 1996.

[12] L. Ljung. *System identification: theory for the user*. Prentice Hall, Upper Saddle River, N.J., 1999.

[13] R. Marino and P. Tomei. *Nonlinear control design: geometric, adaptive and robust*. Prentice Hall International (UK) Ltd., 1996.

[14] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[15] M. Milanese, I. Gerlero, C. Novara, G. Conte, M. Cisternino, C. Pedicini, V. Alfieri, and S. Mosca. Nonlinear mimo data-driven control design for the air and charging systems of diesel engines. In *SAE - ICE2015 - 12th International Conference on Engines and Vehicles*, 2015.

[16] M. Milanese and C. Novara. Set membership identification of nonlinear systems. *Automatica*, 40/6:957–975, 2004.

[17] C. Novara. Sparse identification of nonlinear functions and parametric set membership optimality analysis. *IEEE Transactions on Automatic Control*, 57(12):3236–3241, 2012.

[18] C. Novara. Polynomial model inversion control: numerical tests and applications. *arXiv*, (1509.01421), 2015.

[19] C. Novara, L. Fagiano, and M. Milanese. Direct feedback control design for nonlinear systems. *Automatica*, 49(4):849–860, 2013.

[20] C. Novara, S. Formentin, S.M. Savaresi, and M. Milanese. Data-driven design of two degree-of-freedom nonlinear controllers: the D2-IBC approach. *Automatica*, 72:19–27, 2016.

[21] C. Novara and M. Karimshoushtari. A data-driven model inversion approach to cancer immunotherapy control. In *55th IEEE Conference on Decision and Control*, Las Vegas (Nevada), USA, 2016.

[22] C. Novara and M. Milanese. Control of nonlinear systems: a model inversion approach. *arXiv*, 2014.

[23] M.M. Polycarpou. Stable adaptive neural control scheme for nonlinear systems. *IEEE Transactions on Automatic Control*, 41(3):447–451, 1996.

[24] Z. Qu. *Robust Control of Nonlinear Uncertain Systems*. Wiley series in nonlinear science, 1998.

[25] Jean-Jacques Slotine and Weiping Li. *Applied Nonlinear Control*. Number 7. 1991.

[26] J.A. Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030 –1051, mar. 2006.

[27] A. Yeşildirek and F.L. Lewis. Feedback linearization using neural networks. *Automatica*, 31(11):1659–1664, 1995.