Learning a nonlinear controller from data: theory, computation and experimental results

L. Fagiano* and C. Novara**

***Abstract*—The problem of learning a nonlinear controller directly from experimental data is considered. It is assumed that an existing, unknown controller, able to stabilize the plant, is available, and that input-output measurements can be collected during closed loop operation. A theoretical analysis shows that the error between the input issued by the existing controller and the input given by the learned one shall have low variability in order to achieve closed loop stability. This result is exploited to derive a ℓ_1-norm regularized learning algorithm that achieves the stability condition for a finite number of data points. The approach is completely based on convex optimization. The presented technique is finally tested in real-world experiments to control the flight of a tethered flexible wing, which is characterized by highly nonlinear, unstable and uncertain dynamics subject to external disturbances.**

I. Introduction

The mainstream approaches to control design for nonlinear systems are model based, i.e. they rely on the assumption that a nominal model describing the system's dynamics is available, eventually with a model of the associated uncertainty and/or disturbance. Examples include nonlinear model predictive control [1], control Lyapunov functions [2], [3] and feedback linearization [4]. In many practical applications, a sufficiently accurate (yet simple) nonlinear model of the plant can be derived from first principles, allowing the control designer to rely on such model based techniques.

If the system dynamics are too complex or poorly understood, however, the derivation of a nonlinear model suitable for control design might become an extremely hard task, hence making model based approaches unusable and motivating research in the field of data driven approaches. Generally speaking, the latter can be described as control design techniques where the feedback controller is derived from measured input/output data, without exploiting any dynamical model of the system under study derived from first principles. The only prior information is the knowledge of the measured variables, which can be used for feedback control, and of the manipulated ones. Data driven techniques can be classified in four main categories: indirect techniques, which employ the data to identify a black-box model to be used for control design, [5], [6], [7], [8], [9], [10]; direct techniques, which aim to obtain the controller (often in the form of an inverse model) directly from the available input-output data, hence avoiding the identification of a model of the system, [11], [12], [13], [14], [15], [16], [17]; IMC (Internal Model Control) techniques, where a model is identified from data and the controller (obtained by some inversion approach) is either identified from data or designed from the model, [18], [19]; reinforcement learning/adaptive techniques, where the controller is tuned using the measured data during the online operations, [9], [20], [21]; the latter techniques can be also extended to fault detection, [22]. In [23], several of the mentioned approaches are described, with useful discussions on their practical advantages and drawbacks.

All of the mentioned data driven approaches assume either that no initial controller is known, or that a pre-stabilizing controller is available. In the first case, the plant is assumed to be stable in order to carry out open loop experiments to collect data, while in the second case closed loop experiments are considered. In both cases, the aim is to derive a controller from input/output data using one of the techniques described above, and the eventual pre-stabilizing controller is somehow irrelevant.

In this work, we consider a different control design problem: we do not aim to derive neither a model of the dynamical system, nor an inverted model, but an approximation of an existing controller. Such an approximated controller is required to stabilize the plant and to achieve satisfactory closed-loop performance.

The motivation for treating this problem is twofold. A first setting is related to the control of systems with strong nonlinearities, time-varying behavior, and subject to disturbances, for which the design of an automatic controller is very challenging, yet an experienced human operator can actually achieve good performance through manual control. This case has been studied in previous contributions in the machine learning community, see e.g. [24]. In particular, approaches based on observing and mimicking a human controller, with the aim to extrapolate a set of rules used by the automatic controller, have been proposed and tried on numerical examples. Although the reported numerical results show good performance of the methods, these previous contributions lack a system-theoretic analysis of the stability properties of the closed loop system, and they typically rely on approaches like neural networks and genetic algorithms, for which it is hard to derive guaranteed properties like convexity of the identification problem or optimality of the obtained solution. A second framework is that of reverse engineering, where

This research has received funding from the California Energy Commission under the EISG grant n. 56983A/10-15 "Autonomous flexible wings for high-altitude wind energy generation", and from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n. PIOF-GA-2009-252284 - Marie Curie project "Innovative Control, Identification and Estimation Methodologies for Sustainable Energy Technologies".

*ABB Switzerland Ltd., Corporate Research, Baden-Dättwil - Switzerland

**Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino - Italy

e-mail addresses: lorenzo.fagiano@ch.abb.com, carlo.novara@polito.it

one wants to learn how an existing feedback control system works in order to study and/or replicate it, in simulations or experiments. Examples where this second scenario applies can be very diverse, ranging from the analysis of competitors' solutions in technological fields like automotive controls, to the study of gene regulatory mechanisms in biological sciences [25].

In the described context, we provide three contributions.
The first one is the derivation of sufficient conditions on the system dynamics, existing controller and learned controller, for which the closed loop system, obtained by employing the latter, results to be stable in an input/output sense.
The second contribution is a learning approach based on ℓ_1-norm regularized approximation, which is specifically designed to meet the above-mentioned stability conditions when **the number of available data points is sufficiently large (but finite)**.
Third, we present the results obtained by applying the approach in real-world experiments, concerned with the problem of designing a controller to achieve stable autonomous flight paths of a tethered wing in a wind flow, to be used for airborne wind energy, see e.g. [26], [27]. This problem is particularly challenging due to the difficulty of deriving a reliable model of the system, the presence of strong nonlinearities and the erratic nature of wind. Yet, a human operator with enough training can achieve the desired trajectories by using a human-machine interface. Hence, this experimental setup falls directly in the first class of settings described above, where one wants to design a controller by learning the behavior of a human operator from experimental data. A movie of the experimental tests of the designed controllers is available online [28].

To the best of our knowledge, this paper is one of the first to set up a systematic stability analysis for direct approaches to control system design, to derive from such analysis a computationally tractable design method with theoretical stability guarantees, finally to successfully implement and apply the method to a non-trivial control problem. Hence, all three aspects of theory, implementation and application of the proposed method are treated. As a matter of fact, a good portion of the mentioned literature does not consider the problem of closed-loop stability from a theoretical standpoint and it reports only numerical simulations of academic examples. We point out that in this paper, the approximation is carried out off-line using data measured on the closed-loop system with the existing controller, and it is then implemented on-line as a new feedback controller. On-line tuning and learning to improve the controller are not considered here, albeit they could be topics for future extensions of the approach.

The paper is organized as follows. The problem formulation is stated in Section II. The theoretical results concerned with closed-loop stability and the learning approach are described in subsection III-A and III-B, respectively. The experimental setup and results are presented in Section IV and conclusions are drawn in Section V.

II. Problem formulation

The setting we consider in this work is the following. A single-input, discrete time, nonlinear dynamical system of interest operates in closed loop with an existing controller. The dynamics of the system and of the controller are not known. The system's input variable $u(t)$, i.e. the controller's output, is known and it can be measured at discrete time instants $t \in \mathbb{Z}$. Moreover, $u \in \mathbb{R}$ is limited in a compact $U = [\underline{u}, \overline{u}]$. The system's output variable $y(t)$, i.e. the controller's input, is not known a priori but the control designer can rely on sensors to acquire measurements of different "candidate" feedback variables, based on her/his intuition and experience with the physical process under study. The output y is assumed to belong to a compact set $Y \subset \mathbb{R}^{n_y}$. After a choice of $y(t)$ has been made, we assume that the controller is a static function of this variable:

$$\begin{aligned} &u(t) = \kappa(y(t)) \\ &\kappa : Y \to U. \end{aligned} \tag{1}$$

Moreover, we assume that a disturbance variable $e_s(t)$ is acting on the dynamical system. The variable e_s accounts for (a) exogenous disturbances and the approximation error induced by choosing the input of the controller to be equal to y. The value of $e_s(t)$ is also assumed to belong to a compact set $E_s \subset \mathbb{R}^{n_e}$. We then assume that the chosen output variable evolves in time as follows:

$$\begin{aligned} &y(t+1) = f(y(t), u(t), e_s(t)) \\ &f : Y \times U \times E_s \to Y. \end{aligned} \tag{2}$$

Remark 1: Equation (2) can be seen as a "generalized state equation". Indeed, if the output $y(t)$ includes the system's state, then equation (2) corresponds to a classical state-space description of nonlinear dynamics. If the state is not directly measured but it is observable from the available input-output measurements, then a pseudo-state $y(t)$ can be constructed by taking a suitable number (typically no less than the number of states) of past measurements, such that equation (2) holds. the latter case, f would include a series of unit delay operators in addition to the nonlinear functions describing the system's dynamics. It is important to remark that, in the present framework, the choice of the output signals to be used as feedback variables is a crucial step of the control design. As discussed above, this choice can be carried out on the basis of the available sensors and of the designer's intuition and experience, see for instance the experimental application presented in Section IV.

□

Let us now introduce three assumptions on the functions f and κ. In the following, we will make use of the function sets $\mathcal{K}$ and $\mathcal{KL}$: we recall that $\mathcal{K}$ is the set of all strictly increasing functions $\alpha : \mathbb{R}^+ \to \mathbb{R}^+$ such that $\alpha(0) = 0$, while $\mathcal{KL}$ is the set of all functions $\beta : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ such that for fixed t, $\beta(x,t) \in \mathcal{K}$, and for fixed x, $\lim_{t\to\infty} \beta(x,t) = 0$.

Assumption 1: The function f is Lipschitz continuous over the compact $Y \times U \times E_s$. In particular, it holds that

$$\begin{aligned} &\exists \gamma_f \in (0, +\infty) : \forall e_s \in E_s, \forall y \in Y, \forall u^1, u^2 \in U, \\ &\|f(y, u^1, e_s) - f(y, u^2, e_s)\| \leq \gamma_f |u^1 - u^2|. \end{aligned} \tag{3}$$

□

Assumption 2: The function κ is Lipschitz continuous over the compact Y. □

Assumptions 1-2 imply that the closed loop system:

$$\begin{aligned} &y(t+1) = g(y(t), e_s(t)) \doteq f(y(t), \kappa(y(t)), e_s(t)) \\ &g : Y \times E_s \to Y \end{aligned} \tag{4}$$

is also described by a Lipschitz continuous function g. In particular, by construction, the function g enjoys the following property, which characterize the sensitivity of the closed loop system with respect to the disturbance term e_s:

$$\begin{aligned} &\exists \gamma_{g,e} \in (0, +\infty) : \forall y \in Y, \forall e_s^1, e_s^2 \in E_s, \\ &\|g(y, e_s^1) - g(g, e_s^2)\| \leq \gamma_{g,e} \|e_s^1 - e_s^2\|. \end{aligned} \tag{5}$$

In (3) and (5), the considered norms depend on the problem at hand; in principle any norm can be used and the choice must not be necessarily the same for the signals y and e_s. In the remainder of the paper, this consideration is tacitly implied, unless otherwise stated, and we use the symbol $\|\cdot\|$ for the sake of simplicity.

Assumptions 1-2 are quite standard in nonlinear control analysis and design and they are reasonable, since in practice the inputs, disturbance and outputs of the process under study are often bounded in some compact sets and the functions describing the system and the controller are assumed to be differentiable on such compact sets, hence Lipschitz continuous.

The dynamical system described by g has e_s as input and y as output. properties of the closed-loop system clearly depend on the controller κ, which is assumed to be stabilizing. In particular, we consider the following stability notion, see e.g. [29]:

Definition 1: A nonlinear system with input e_s and output y, is finite-gain ℓ_∞ stable if a function $\alpha \in \mathcal{K}$, a function $\beta \in \mathcal{KL}$ and a scalar $\delta > 0$ exist, such that:

$$\forall t \geq 0,\ \|y(t)\| \leq \alpha(\|\mathbf{e}_s\|_\infty) + \beta(\|y(0)\|, t) + \delta. \tag{6}$$

□

In Definition 1, the generic signal $\mathbf{v} \doteq \{v(0), v(1), ...\}$ is given by the infinite sequence of values of the variable $v(t)$, $t \geq 0$, and $\|\mathbf{v}\|_\infty \doteq \max_{t \geq 0} \|v(t)\|$ is the $\ell_\infty-$norm of the signal $\mathbf{v}$ with the underlying norm taken to be the vector norm $\|v\|$.

The stabilizing properties of κ are formalized by the following assumption:

Assumption 3: The functions κ and f are such that the following property holds:

$$\forall t \geq 0,\ \|g(y(t), 0)\| \leq \gamma_{g,y} \|y(t)\|,\ \gamma_{g,y} \in (0, 1). \tag{7}$$

□

Assumption 3 implies that the closed-loop system (4) enjoys finite-gain ℓ_∞ stability as given in Definition 1, in particular with straightforward passages we have:

$$\forall t \geq 0,\ \|y(t)\| \leq \underbrace{\frac{\gamma_{g,e}}{1 - \gamma_{g,y}} \|\mathbf{e}_s\|_\infty}_{\alpha(\|\mathbf{e}_s\|_\infty)} + \underbrace{\gamma_{g,y}^t \|y(0)\|}_{\beta(\|y(0)\|, t)}, \tag{8}$$

see the Appendix for a derivation of this inequality.

Remark 2: Assumption 3 basically requires that the norm $\|y\|$ is a Lyapunov function for the closed-loop system, with $y = 0$ being the asymptotically stable equilibrium for the unperturbed system. This assumption is reasonable, since the choice of y is a design aspect in the approach proposed here, and we point out that many cases occurring in practice can be treated in the considered settings. For example, weighted norms of the output signals, energy functions, nonlinear Lyapunov functions and trajectory tracking problems can be treated by suitably defining the variable y. □

Overall, Assumptions 1-3 are reasonable (and quite common) in the context of system identification, function approximation and learning, since a stable system is needed to collect data and carry out identification experiments. In particular, in this work we will consider a finite number N of input and output measurements, indicated as $\tilde{u}(t)$, $\tilde{y}(t)$, $t = -N, \ldots, -1$, collected from the system operating in closed loop with the unknown controller κ. These data points are assumed to be affected by additive noise variables, indicated as $e_u(t)$ and $e_y(t)$, respectively:

$$\begin{aligned} \tilde{u}(t) &= u(t) + e_u(t) \\ \tilde{y}(t) &= y(t) + e_y(t). \end{aligned} \tag{9}$$

Note that $e_u(t)$ may include both measurement noise and errors arising in the application of the control law. The latter can be present for example if the aim is to learn a controller from the behavior of a human operator, who might be subject to fatigue and mistakes.

The noise variables are assumed to satisfy the following boundedness properties where, for a generic variable $q \in \mathbb{R}^{n_q}$ and scalar $\rho \in (0, +\infty)$, we denote the n_q-dimensional norm ball set of radius ρ as $B_\rho \doteq \{q \in \mathbb{R}^{n_q} : \|q\| \leq \rho\}$:

Assumption 4: The following boundedness properties hold:

(a) $e_u(t) \in B_{\varepsilon_u},\ \forall t \geq 0$;

(b) $e_y(t) \in B_{\varepsilon_y},\ \forall t \geq 0$. □

According to (1), with straightforward manipulations, the measured data can be described by the following set of equations:

$$\tilde{u}(t) = \kappa(\tilde{y}(t)) + d\,(t)\,,\ t = -N, \ldots, -1$$

where $d\,(t)$ accounts for the noises $e_u(t)$ and $e_y(t)$ in (9). Since $e_u(t)$ and $e_y(t)$ are bounded and κ is Lipschitz continuous, it follows that $d\,(t)$ is also bounded:

$$d\,(t) \in B_\varepsilon,\ \forall t \geq 0. \tag{10}$$

The following assumption on the pair $(\tilde{y}(t), d\,(t))$ is considered.

Assumption 5: The set of points $\mathcal{D}_{yd}^N \doteq \{(\tilde{y}(t), d\,(t))\}_{t=-N}^{-1}$ is dense on $Y \times B_\varepsilon$ as $N \to \infty$. That is, for any $(y, d) \in Y \times B_\varepsilon$ and any $\lambda \in \mathbb{R}^+$, a finite $N_\lambda \in \mathbb{N}$ and a pair $(\tilde{y}(t), d\,(t)) \in \mathcal{D}_{yd}^{N_\lambda}$ exist such that $\|(y, d) - (\tilde{y}(t), d\,(t))\| \leq \lambda$. □

Assumption 5 essentially ensures that the controller domain Y is "well explored" by the data $\tilde{y}(t)$ and, at the same time, the noise $d(t)$ covers its domain B_ε, hitting the bounds $-\varepsilon$ and ε with arbitrary closeness after a sufficiently long time. This noise property is called tightness, see [16] and, for a probabilistic version, [30], and it is somehow linked to a persistence of excitation condition when the data points are collected.

In the described setting, the problem we want to address can be stated as follows:

Problem 1: learn a controller $\hat{\kappa}$ from N measurements $\tilde{y}$ and $\tilde{u}$, obtained from the system operating in closed-loop with an unknown controller κ, such that:

1) **a finite N exists such that $\hat{\kappa}$ renders the closed loop system finite-gain ℓ_∞ stable;**
2) the trajectory deviation induced by the use of $\hat{\kappa}$ instead of κ is "small";
3) $\hat{\kappa}$ has "low" complexity, to be easily implementable on real-time processors. □

Remark 3: The control design problem considered here is different from the one considered in [16], where the aim is to track reference sequences belonging to a certain signal set. As a consequence, the stability analysis presented in the following is significantly different from the one in [16]. As discussed in Remark 5 below, also the control design algorithms presented in the two papers are quite different. □

III. Learning a controller from data: theory and computation

In this section, we present an approach that is able to solve **Problem** 1. In order to do so, we first derive a sufficient condition for a generic controller $\hat{\kappa} \approx \kappa$ to stabilize the closed-loop system and then we propose a technique, based on convex optimization, that is able to learn a controller $\hat{\kappa}$ **which enjoys the derived stability condition for a sufficiently large** N.

A. Closed loop stability analysis

Our first aim is to derive a sufficient condition on the controller $\hat{\kappa}$, such that the obtained closed loop system is finite-gain ℓ_∞ stable. The controller $\hat{\kappa}$ is chosen to be Lipschitz continuous over the compact Y, with constant $\gamma_{\hat{\kappa}}$:

$$\begin{array}{l}\exists \gamma_{\hat{\kappa}} \in (0, +\infty) : \forall y^1, y^2 \in Y, \\ |\hat{\kappa}(y^1) - \hat{\kappa}(y^2)| \leq \gamma_{\hat{\kappa}} \|y^1 - y^2\|.\end{array} \tag{11}$$

Let us define the error function $\Delta : Y \to \mathbb{R}$:

$$\Delta(y) \doteq \kappa(y) - \hat{\kappa}(y). \tag{12}$$

We denote with $\Delta_0 \doteq \Delta(0)$ the error function evaluated at $y = 0$. By construction, the error function is Lipschitz continuous, with some constant $\gamma_\Delta \in (0, +\infty)$:

$$\begin{array}{l}\exists \gamma_{\Delta} \in (0, +\infty) : \forall y^1, y^2 \in Y, \\ |\Delta(y^1) - \Delta(y^2)| \leq \gamma_{\Delta} \|y^1 - y^2\|.\end{array} \tag{13}$$

We indicate with $\hat{g}$ the closed loop system obtained by using the controller $\hat{\kappa}$. In particular, $\hat{g}$ is defined as

$$\begin{array}{l}y(t+1) = \hat{g}(y(t), e_s(t), e_y(t)) \doteq f(y(t), \hat{\kappa}(y(t) + \\ e_y(t)), e_s(t)),\ \hat{g} : Y \times E \times B_{\varepsilon_y} \to Y.\end{array} \tag{14}$$

Note that the feedback variable used by the learned controller $\hat{\kappa}$ is the noise-corrupted measurement of the output y. The next result provides a sufficient condition for the controller $\hat{\kappa}$ to stabilize the closed loop system.

Theorem 1: Let Assumptions 1-3 and 4-(b) hold. If

$$\gamma_\Delta < \frac{1 - \gamma_{g,y}}{\gamma_f}, \tag{15}$$

then the closed-loop system $\hat{g}$ is finite-gain ℓ_∞ stable. More precisely, it holds that

$$\begin{array}{l}\forall t \geq 0,\ \|y(t)\| \leq \underbrace{\frac{\gamma_{g,e}}{1-\gamma}\|\mathbf{e}_s\|_\infty}_{\alpha(\|\mathbf{e}_s\|_\infty)} + \underbrace{\gamma^t \|y(0)\|}_{\beta(\|y(0)\|,t)} + \\ \underbrace{\frac{1}{1-\gamma}\left(\gamma_f |\Delta_0| + \gamma_f \gamma_{\hat{\kappa}} \varepsilon_y\right)}_{\delta},\end{array} \tag{16}$$

with $\gamma \doteq (\gamma_\Delta \gamma_f + \gamma_{g,y}) < 1$.

Proof. See the Appendix. □

It is worth commenting on the result of Theorem 1. Roughly speaking, γ_Δ gives an indication on the regularity of the error function $\Delta = \kappa - \hat{\kappa}$. Assuming for example that Δ is differentiable, a low γ_Δ means that the quantity $\|d\Delta/dy\|$ is bounded by a small value, i.e. the variability of the error over the set Y is low. This happens e.g. when the functions κ and $\hat{\kappa}$ differ by some offset, but have similar shapes. A large value of γ_Δ, on the other hand, indicates that the control error can have high variability, as it can happen e.g. when the controller $\hat{\kappa}$ is over-fitting the available measured data. Theorem 1 states that the quantity γ_Δ should be sufficiently small in order to guarantee closed-loop stability, and how small depends on the features of the plant to be controlled and of the unknown controller κ. In particular, the more "sensitive" is the plant to input perturbation, i.e. the larger is the Lipschitz constant γ_f, and the worse are the stabilizing properties of the controller κ, i.e. the closer is the constant $\gamma_{g,y}$ in (7) to 1, the smaller γ_Δ has to be in order to meet the sufficient condition. In other words, the Theorem indicates that the quality of the learned controller $\hat{\kappa}$, in terms of low variability of the control error, should be higher if the uncontrolled system is more sensitive to input perturbations and/or if the closed-loop system obtained with κ is closer to being unstable.

The value of γ_Δ influences also the decay rate of the term related to the initial condition $\|y(0)\|$, consider eq. (16), as well as the gain in the additive term δ. As to the latter, a comparison with the analogous term in (8) reveals the effects of the absolute value of the control error and of the presence of output noise. The former is represented by the quantity $|\Delta_0|$, i.e. the magnitude of the control error evaluated at $y = 0$. Note that the choice of $y = 0$ to evaluate this term is not restrictive, since a simple coordinate change can be used to refer all the results to a different output value. According to the result, the smaller the value of $|\Delta_0|$, the closer is the term δ to the one obtained with the unknown controller κ, i.e. 0. This aspect, coupled with the condition (15) on the value of γ_Δ, basically states that, in order to better replicate the behavior obtained with the controller κ, the control error function has to be small in absolute value and have low variability, as the intuition would suggest. About the noise term, it contributes to δ in (16) in a way proportional to its maximum norm ε_y, and the gain depends on how sensitive the controller $\hat{\kappa}$ is to perturbations of its input argument, as indicated by the value

of $\gamma_{\hat{\kappa}}$. Since $\hat{\kappa}$ is an approximation of κ, it can be expected that the two functions will have similar Lipschitz constants, hence a large constant γ_κ of the true controller would lead to a large constant also of the approximated ones, i.e. to higher sensitivity to measurement noise and to generally worse performance. Finally, note that the effects of $|\Delta_0|$ and ε_y are proportional to γ_f, i.e. to how sensitive the uncontrolled plant is to input perturbations, and inversely proportional to $1-\gamma$, i.e. to how close the closed loop system is to being unstable in the sense of Definition 1.

The results presented so far serve as a theoretical justification of the learning algorithm that we present in the next section, which indeed is able to satisfy condition (15) in the limit, hence providing a solution to **Problem** 1.

B. Learning algorithm

A parametric representation is considered for the controller $\hat{\kappa}$:

$$\hat{\kappa}(y) = \sum_{i=1}^{M} \hat{a}_i \varphi_i(y) \tag{17}$$

where $\varphi_i : Y \to U$ are Lipschitz continuous basis functions. The coefficients $\hat{a}_i \in \mathbb{R}$ are identified by means of the following Algorithm 1.

Algorithm 1 1: Controller learning.

1) Take a set of basis functions $\{\varphi_i\}_{i=1}^{M}$. The choice of this set can be carried out following the indication in Section III-C.
2) Using the data set $\mathcal{D}^N \doteq \{\tilde{u}(t), \tilde{y}(t)\}_{t=-N}^{-1}$ and the basis functions chosen at step 1), define the following quantities:

$$\Phi \doteq \begin{bmatrix} \varphi_1(\widetilde{y}(-N)) & \cdots & \varphi_M(\widetilde{y}(-N)) \\ \vdots & \ddots & \vdots \\ \varphi_1(\widetilde{y}(-1)) & \cdots & \varphi_M(\widetilde{y}(-1)) \end{bmatrix} \in \mathbb{R}^{N\times M}$$
$$\widetilde{\boldsymbol{u}} \doteq (\widetilde{u}(-N), \ldots, \widetilde{u}(-1)) \in \mathbb{R}^{N\times 1}.$$

3) Using Algorithms 2 and 3 below, obtain an estimate $\hat{\varepsilon}$ of the noise bound ε in (10), and estimates $\hat{\gamma}_f$ and $\hat{\gamma}_{g,y}$ of the constants γ_f and $\gamma_{g,y}$ in (3) and (7). Choose $\gamma'_\Delta = \bar{\gamma} - \varepsilon_s$, where $\bar{\gamma} \doteq (1-\hat{\gamma}_{g,y})/\hat{\gamma}_f$ and $\varepsilon_s > 0$ is an arbitrarily small constant.
4) Solve the following convex optimization problem:

$$\begin{aligned} &a^1 = \arg\min_{a\in\mathbb{R}^M} \|a\|_1 \\ &\text{subject to} \\ &(a)\ \ \|\widetilde{\boldsymbol{u}} - \Phi a\|_\infty \le \alpha\hat{\varepsilon} \\ &(b)\ \ |\widetilde{u}(k) - \widetilde{u}(t) + (\Phi_t^r - \Phi_k^r)\, a| \\ &\le \gamma'_\Delta \|\widetilde{y}(k) - \widetilde{y}(t)\| + 2\hat{\varepsilon},\ \begin{cases} k = -N, \ldots, -1 \\ t = k+1, \ldots, -1 \end{cases} \end{aligned} \tag{18}$$

where $\Phi_t^r \doteq [\ \varphi_1(\widetilde{y}(t)) \ \cdots \ \varphi_M(\widetilde{y}(t))\ ]$ and $\alpha \ge 1$ is a number slightly larger than the minimum value for which the constraint (a) is feasible.

5) Obtain the coefficient vector $\hat{a} = (\hat{a}_1, \ldots, \hat{a}_M)$ by solving the following convex optimization problem:

$$\begin{aligned} &(\hat{a}, \gamma_\Delta^s) = \arg\min_{a\in\mathbb{R}^M,\, \gamma''_\Delta\in\mathbb{R}^+} \gamma''_\Delta \\ &\text{subject to} \\ &(a)\ \ \|\widetilde{\boldsymbol{u}} - \Phi a\|_\infty \le \alpha\hat{\varepsilon} \\ &(b)\ \ |\widetilde{u}(k) - \widetilde{u}(t) + (\Phi_t^r - \Phi_k^r)\, a| \\ &\le \gamma''_\Delta \|\widetilde{y}(k) - \widetilde{y}(t)\| + 2\hat{\varepsilon},\ \begin{cases} k = -N, \ldots, -1 \\ t = k+1, \ldots, -1 \end{cases} \\ &(c)\ \ a_i = 0,\ \forall i \notin \text{supp}\left(a^1\right) \end{aligned} \tag{19}$$

where $\text{supp}\left(a^1\right)$ is the support of a^1, i.e. the set of indices at which a^1 is not null. □

The rationale behind the algorithm can be explained as follows. After the preliminary operations carried out in steps 1)-3), the ℓ_1 norm of the coefficient vector a is minimized in step 4), leading to a sparse coefficient vector a^1, i.e. a vector with a "small" number of non-zero elements. Constraint (a) in (18) ensures the consistency between the measured data and the prior information on the noise affecting these data (assuming that $\hat{\varepsilon}$ is a reliable estimate of ε and α is close to 1). Constraints (b) allow us to guarantee closed-loop stability **when a sufficiently large number of data is used**, see Theorem 4 below. As it is well known from previous results [31], [32], [33], [34], [35], [36], minimizing the ℓ_1 norm is useful not only for sparsity but also for basis function selection. In this view, step 4 allows us to select those basis functions which are more appropriate for ensuring stability.

Step 5) aims at reducing the Lipschitz constant of the error function, maintaining the same accuracy and sparsity level obtained in step 4), and satisfying the constraints for closed-loop stability. Indeed, reducing this constant allows us to satisfy the stability condition (15) with a larger margin. step 5) of the algorithm accounts for the requirement 2) of **Problem** 1.

The reason why a sparse controller is looked for is twofold. First, a sparse function is easy to implement on real-time processors, which may have limited memory and computational capacity, hence accounting for the requirement 3) of **Problem** 1. Second, sparse functions have good regularity properties and are thus able to provide good accuracy on new data by limiting well-known issues such as over-fitting and the curse of dimensionality. Among the existing approaches in sparse identification (see e.g. [31], [32], [33], [34], [35], [36]), ℓ_1 norm relaxation algorithms replace the minimization of the ℓ_0 quasi-norm of the vector of coefficients with the minimization of its ℓ_1 norm. 1 is essentially a modified ℓ_1 relaxation method: in step 4), an optimization problem is solved, where the ℓ_0 quasi-norm is replaced by the ℓ_1 norm, and additional constraints for closed-loop stability are used (i.e. (b) in (18)). In step 5), a vector $\hat{a}$ is obtained, with the same support as a^1, which minimizes the estimated Lipschitz constant of the error function and satisfies the closed-loop stability condition evaluated on the available data. We note that in principle a significant number of other identification methods from the literature could be employed for control design, see [37], [38] and the references therein. However, these methods appear to be not very suitable for imposing controller sparsity and

closed-loop stability (except perhaps support vector machines, [38], which anyway have connections with sparse methods, [39]).

Remark 4: If a small number of data is used for control design, it may happen that $(1-\hat{\gamma}_{g,y})/\hat{\gamma}_f \leq 0$, thus not allowing a feasible choice of the Lipschitz constant γ'_Δ in step 3 of Algorithm 1. In this case, our indication is to collect a larger number of data in order to let the estimated Lipschitz constants $\hat{\gamma}_f$ and $\hat{\gamma}_{g,y}$ get closer to the true values, which by assumption satisfy the condition $(1-\gamma_{g,y})/\gamma_f \leq 0$. If collecting more data is not possible, our indication is to choose γ'_Δ slightly larger than the minimum value for which the optimization problem (18) is feasible (with a not too large α). Similar indications hold for the case where $(1-\hat{\gamma}_{g,y})/\hat{\gamma}_f > 0$ but the chosen γ'_Δ is too small and constraint (b) in (18) is thus not feasible. □

Remark 5: Algorithm 1 is different from Algorithm 1 in [16], for three main features: 1) Algorithm 1 in [16] accounts for the presence of a reference signal in the controller, see also Remark 3; 2) differently from Algorithm 1 in [16], the parameters required by the present Algorithm 1 are chosen/estimated automatically (see Subsection III-C below); 3) Algorithm 1 in [16] does not minimize the Lipschitz constant of the error function. Reducing this constant is clearly important in order to increase the "stability margin". While feature 1) has limited relevance, since it derives from the difference of the control design problem considered here from the one considered in [16], features 2) and 3) represent a significant improvement with respect to the approach in [16]. □

C. Parameter estimation and basis function choice

All the parameters involved in Algorithm 1 (i.e. the noise bound $\hat{\varepsilon}$ and the constants $\hat{\gamma}_f$ and $\hat{\gamma}_{g,y}$) can be estimated in a systematic way by means of the following Algorithms.

Suppose that a set of data $\{\widetilde{w}(t), \widetilde{z}(t)\}_{t=-N}^{-1}$ is available, described by

$$\widetilde{z}(t) = \mathfrak{f}(\widetilde{w}(t)) + e(t),\ t = -N, \ldots, -1 \tag{20}$$

where $\mathfrak{f}: W \to \mathbb{R}$ is a generic unknown function, $W \subset \mathbb{R}^{n_w}$ and $e(t)$ is an unknown noise. Assume that $e(t) \in B_\varepsilon$, $\forall t$, and $\mathfrak{f}$ is Lipschitz continuous with constant $\gamma_\mathfrak{f}$. The noise bound ε and the Lipschitz constant $\gamma_\mathfrak{f}$ can be estimated as follows.

Algorithm 1 2: Noise bound estimation.

1) Choose a "small" $\rho > 0$. For example: $\rho = 0.01 \max_{t,k=-N,\ldots,-1} \|\widetilde{w}(t) - \widetilde{w}(k)\|$.
2) For $t = -N, \ldots, -1$, compute
$$\delta\tilde{z}_t = \max_{i,j \in J_t} |\widetilde{z}(i) - \widetilde{z}(j)|$$
where $J_t \doteq \{k : \|\widetilde{w}(t) - \widetilde{w}(k)\| \leq \rho\}$. If $J_t = \emptyset$, set $\delta\tilde{z}_t = \infty$. If $J_t = \emptyset$ for all $t = -N, \ldots, -1$, go to step 1) and choose a larger ρ.
3) Obtain the estimate $\hat{\varepsilon}$ of the noise bound ε as
$$\hat{\varepsilon} = \frac{1}{2\hat{N}} \sum_{t \in Q} \delta\tilde{z}_t$$
where $Q \doteq \{t \in \{-N, \ldots, -1\} : \delta\tilde{z}_t < \infty\}$ and $\hat{N} \doteq \text{card}\,(Q)$.

□

Algorithm 1 3: Lipschitz constant estimation.

1) For $t, k = -N, \ldots, -1$ and $\widetilde{w}(t) \neq \widetilde{w}(k)$, compute
$$\tilde{\gamma}_{kt} = \begin{cases} \frac{\widetilde{z}(t) - \widetilde{z}(k) - 2\hat{\varepsilon}}{\|\widetilde{w}(t) - \widetilde{w}(k)\|}, & \text{if } \widetilde{z}(t) > \widetilde{z}(k) + 2\hat{\varepsilon} \\ \frac{\widetilde{z}(k) - \widetilde{z}(t) - 2\hat{\varepsilon}}{\|\widetilde{w}(t) - \widetilde{w}(k)\|}, & \text{if } \widetilde{z}(k) > \widetilde{z}(t) + 2\hat{\varepsilon} \\ 0, & \text{otherwise} \end{cases}$$
where $\hat{\varepsilon}$ is the noise bound estimated by Algorithm 2.
2) Obtain the estimate $\hat{\gamma}$ of the Lipschitz constant $\gamma_\mathfrak{f}$ as
$$\hat{\gamma} = \max_{t,k=-N,\ldots,-1 : \widetilde{w}(t) \neq \widetilde{w}(k)} \tilde{\gamma}_{kt}$$

□

The two algorithms above allow one to estimate the Lipschitz constant of a generic function $\mathfrak{f}$. We now discuss how the algorithms can be applied to estimate the constants γ_f and $\gamma_{g,y}$, which are required by the learning algorithm 1.

The Lipschitz constant γ_f of f with respect to $u(t)$ can be estimated considering that

$$\tilde{y}(t+1) = \mathfrak{f}(\tilde{u}(t)) + v_f(t),\ t = -N, \ldots, -1$$

where $\mathfrak{f}(\tilde{u}(t)) \doteq f(y^*, \tilde{u}(t), e_s^*)$ is an unknown function with Lipschitz constant γ_f, the quantities y^* and e_s^* are defined as

$$(y^*, e_s^*) = \arg\max_{(y,e) \in Y \times E_s} \mathfrak{L}_f(y, e)$$
$$\mathfrak{L}_f(y, e) \doteq \max_{u^1, u^2 \in U} \frac{\|f(y, u^1, e) - f(y, u^2, e)\|}{|u^1 - u^2|}$$

and

$$v_f(t) \doteq f(y(t), u(t), e_s(t)) - f(y^*, \tilde{u}(t), e_s^*) + e_y(t+1) \tag{21}$$

is an unknown noise. Note that $v_f(t)$ is bounded, due to the boundedness of $u(t)$, $e_s(t)$ and $y(t)$ and the Lipschitz continuity of f: $v_f(t) \in B_{\varepsilon_f}$, $\forall t \geq 0$. A bound on $v_f(t)$ can be estimated by means of Algorithm 2. Then, Algorithm 3 can be applied to estimate γ_f, posing $\widetilde{z}(t) = \tilde{y}(t+1)$ and $\widetilde{w}(t) = \tilde{u}(t)$.

The constant $\gamma_{g,y}$ can be estimated analogously. Indeed, according to (7), $\gamma_{g,y}$ is the rate of change of g with respect to $y(t)$ on the system trajectory. Hence, we can consider that

$$\tilde{y}(t+1) = \mathfrak{g}(\tilde{y}(t)) + v_g(t),\ t = -N, \ldots, -1$$

where $\mathfrak{g}(\tilde{y}(t)) \doteq g(\tilde{y}(t), e_s^*)$ is an unknown function and quantity e_s^* is defined as

$$e_s^* = \arg\max_{e \in E_s} \mathfrak{L}_g(e),\ \mathfrak{L}_g(e) \doteq \max_{y^1, y^2 \in Y} \frac{\|g(y^1, e) - g(y^2, e)\|}{|y^1 - y^2|}$$

and

$$v_g(t) \doteq g(y(t), e_s(t)) - g(\tilde{y}(t), e_s^*) + e_y(t+1) \tag{22}$$

is an unknown noise. Note that $v_g(t)$ is bounded, due to the boundedness of $e_s(t)$ and $y(t)$ and the Lipschitz continuity of g: $v_g(t) \in B_{\varepsilon_g}$, $\forall t \geq 0$. A bound on $v_g(t)$ can be estimated by means of Algorithm 2. Then, Algorithm 3 can be applied to estimate $\gamma_{g,y}$, posing $\widetilde{z}(t) = \tilde{y}(t+1)$, $\widetilde{w}(t) = \tilde{y}(t)$ and $k = t - 1$.

Another important step of Algorithm 1 is the choice of the basis functions φ_i (this aspect is crucial for any identification

method relying on a basis function representation, [40], [41]). An inappropriately chosen family of functions can force the retention of many terms by the identification algorithm or can lead to large approximation errors. In these situations, several problems may arise, such as high controller complexity, closed-loop instability and/or large deviations from the ideal trajectory. The following indications can be considered to address this issue.

The first step is to run Algorithm 1 using a given family of basis functions (e.g. Gaussian, sigmoidal, wavelet, polynomial, trigonometric).

Then, the following cases can occur:

(a) α is small (close to 1) and $\hat{a}$ is sparse: the chosen basis functions are adequate, since a small number of them is able to explain the measured data.

(b) α is small and $\hat{a}$ is not sparse: the basis function choice is not appropriate. Indeed, using a large number of basis functions may lead to overfitting problems and possibly to closed-loop instability. Then, one should choose a different family of basis functions.

(c) α is not small: the basis function choice is not appropriate since it leads to a large approximation error on the measured data. Also in this case, one should choose a different family of basis functions.

The quality of the derived approximation can be also assessed by testing it on data that were not used in the learning algorithm, as it is commonly done in identification problems.

D. Theoretical guarantees

In this subsection, the theoretical properties of Algorithms 1, 2 and 3 are analyzed.

The first result is that the noise bound $\hat{\varepsilon}$ estimated by Algorithm 2 converges to its true value when the number N of data tends to infinity.

Theorem 2: Let the set $\{(\widetilde{w}(t), e(t))\}_{t=-N}^{-1}$ appearing in (20) be dense on $W \times B_\varepsilon$ as $N \to \infty$. Then,

$$\lim_{N\to\infty} \hat{\varepsilon} = \varepsilon$$

where $\hat{\varepsilon}$ is the noise bound estimated by Algorithm 2.

Proof. See the Appendix. □

The estimate of ε is used by Algorithm 3 to provide an estimate of the constant $\hat{\gamma}$. The second result is that this estimate also converges to its true value as $N \to \infty$.

Theorem 3: Let the set $\{(\widetilde{w}(t), e(t))\}_{t=-N}^{-1}$ appearing in (20) be dense on $W \times B_\varepsilon$ as $N \to \infty$. Then,

$$\lim_{N\to\infty} \hat{\gamma} = \gamma_{\mathfrak{f}}$$

where $\hat{\gamma}$ is the constant estimated by Algorithm 3.

Proof. See the Appendix. □

A third result is now presented, based on Theorems 2 and 3, showing that the controller $\hat{\kappa}$ identified by means of Algorithm 1 satisfies the stability condition (15) as $N \to \infty$. Before stating this result, let us introduce two technical assumptions, regarding the noises $v_f(t)$ and $v_g(t)$ defined in (21) and (22), respectively.

Assumption 6: The set of points $\mathcal{D}_{uv}^N \doteq \{\widetilde{u}(t), v_f(t)\}_{t=-N}^{-1}$ is dense on $U \times B_{\varepsilon_f}$ as $N \to \infty$.

□

Assumption 7: The set of points $\mathcal{D}_{yv}^N \doteq \{(\widetilde{y}(t), v_g(t))\}_{t=-N}^{-1}$ is dense on $Y \times B_{\varepsilon_g}$ as $N \to \infty$.

□

We now have all the ingredients needed to derive our main result on the stabilizing properties of the learned controller.

Theorem 4: **Let the optimization problem (18) be feasible for any $N \geq 0$. Let Assumptions 1-2, 4, and 5-7 hold. Then, there exists a finite N such that the closed-loop control system with the learned controller $\hat{\kappa}$ is finite-gain ℓ_∞ stable.**

Proof. See the Appendix. □

In light of Theorem 1, Theorem 4 implies that the closed loop system obtained by using the controller $\hat{\kappa}$ is, in the limit, finite-gain ℓ_∞ stable according to Definition 1. We note that the regularity constraints (b) in (18) are essential to achieve this result. In fact, in the absence of such constraints the approach would simply try to find the set of coefficients with minimal ℓ_1-norm such that the control inputs given by the approximated controller are consistent with the experimental data and the estimated disturbance/noise bounds. The resulting controller would not enjoy the same guarantees, unless some other additional assumptions are considered. One such assumption would be that the unknown controller κ belongs to the set of approximated controllers that can be obtained with the chosen parametrization, i.e. that there exist some value of the coefficient vector a^* such that $\kappa(y) = \sum_{i=1}^{M} a_i^* \varphi_i(y)$. However this would be a quite restrictive and unrealistic requirement. The method we proposed is able to provide stabilizing properties without the need for such additional assumptions.

We illustrate now the convergence result of Theorem 4 through a simple numerical example.

Example: Estimated Lipschitz constant.

We have considered the function

$$u = \kappa(y) \doteq 2ye^{-y^2}\cos(8y),$$

shown in Fig. 1, and values of $N = 10, 20, \ldots, 250$. For each one of these values, we generated a data set $\mathcal{D}^N \doteq \{\tilde{u}(t), \tilde{y}(t)\}_{t=-N}^{-1}$ according to

$$\widetilde{u}(t) = \kappa(\widetilde{y}(t)) + d(t),\ t = -N, \ldots, -1$$

where $d(t)$ is a white uniform noise with amplitude 0.05. Then, we applied Algorithm 1 to obtain an estimate $\hat{\kappa}^N$ of κ of the form (17), where $\varphi_i : [-3, 3] \to [-1, 1]$ are Gaussian basis functions:

$$\varphi_i(y) = e^{-100(y-c(i))^2},\ i = 1, \ldots, 100$$

where the centers $c(i)$ are uniformly distributed in the interval $[-3, 3]$. The motivation for using Gaussian functions is that they are universal approximators, see e.g. [42]. Note that, thanks to its sparsification properties, the algorithm selected in average 20 basis functions (the number of selected functions did not change significantly with N).

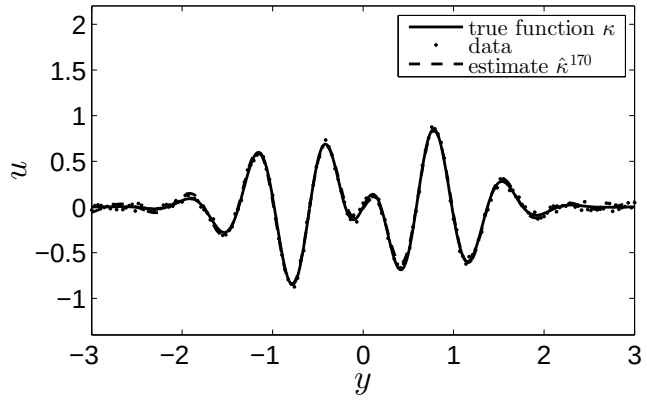

Fig. 1. Numerical example on the estimated Lipschitz constant. True function, data, and estimate $\hat{\kappa}^{170}$ obtained by using $N = 170$ data points.

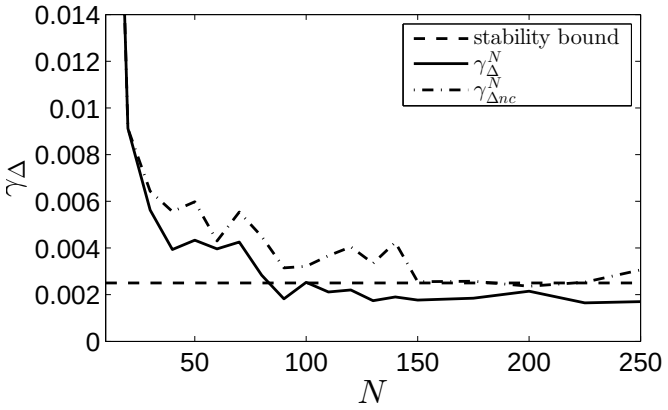

Fig. 2. Numerical example on the estimated Lipschitz constant. Course of the Lipschitz constants γ_Δ^N, obtained by using the regularity constraints (b) in (18), and $\gamma_{\Delta nc}^N$ obtained without using such constraints. It can be noted that with a sufficiently large number of data points the value of γ_Δ^N is systematically below the required threshold, while that of $\gamma_{\Delta nc}^N$ is above.

For comparison, we computed another estimate $\hat{\kappa}_{nc}^N$ of the same form (17), with the same basis functions. This estimate has been obtained by means of Algorithm 1, without the constraints (b) in (18). We recall that, according to Theorem 4, these constraints ensure the convergence of the Lipschitz constant of the error function to a value yielding closed-loop stability.

The Lipschitz constant γ_Δ^N of the error function $\Delta^N \doteq \kappa - \hat{\kappa}^N$ and the Lipschitz constant $\gamma_{\Delta nc}^N$ of the error function $\Delta^N \doteq \kappa - \hat{\kappa}_{nc}^N$ are shown in Fig. 2 for $N = 10, 20, \ldots, 250$. It can be noted that γ_Δ^N decreases quite rapidly as N becomes large, quickly taking values below an hypothetical threshold that is sufficient for closed-loop stability. Also $\gamma_{\Delta nc}^N$ seems to have a similar behavior, however the decrease is slower and less regular with respect to the one of γ_Δ^N and, in any case, satisfaction of the stability condition is not ensured theoretically. In Fig. 1, the estimate $\hat{\kappa}^{170}$ is compared with the true function κ.

IV. Experimental application: automatic crosswind flight of tethered flexible wings

We present an application of the control learning approach in real-world experiments, with a small-scale prototype built at the University of California, Santa Barbara to study the dynamics and control of tethered wings. This research is related to airborne wind energy technologies, e.g. [26], [27], [43], [44] for details.

A. System description and control objective

We consider a flexible wing connected by three lines to a ground unit (GU).

The wing's trajectory evolves downwind with respect to the GU. We define an inertial coordinate system $G \doteq (X, Y, Z)$ centered at the Gu location and with the X axis aligned with the average wind direction. considering a fixed length of the lines, denoted by r, the wing's trajectory is confined on a quarter sphere, commonly named "wind window", see Fig. 3 (dashed lines).

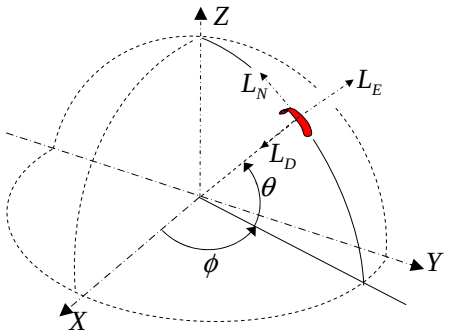

Fig. 3. Reference system $G = (X, Y, Z)$, wind window (dashed lines), variables θ, ϕ, and local north, east and down (L_N, L_E, L_D) axes.

The wing's position $\vec{p}(t)$ can be expressed in the inertial frame G by using the spherical coordinates $\theta(t)$, $\phi(t)$ (see Fig. 3), where t is the continuous time variable. We also consider a non-inertial coordinate system, $L \doteq (L_N, L_E, L_D)$, centered at the wing's position (depicted in Fig. 3). The L_N axis, or local north, is tangent to the sphere of radius r, on which the wing's trajectory evolves, and points towards its zenith. The L_D axis, called local down, points the center of the sphere (i.e. the GU), hence it is perpendicular to the tangent plane to the sphere at the wing's location. The L_E axis, named local east, forms a right-handed system and spans the tangent plane together with L_N. The system L is a function of the wing's position only, see [45] for a complete derivation. The wing velocity vector can be expressed in the L frame as

$$\frac{d}{dt}\vec{p}(t) = r \begin{pmatrix} \dot{\theta} \\ \cos(\theta)\dot{\phi} \\ 0 \end{pmatrix}. \tag{23}$$

Finally, the velocity angle $\gamma(t)$ of the wing is defined as:

$$\gamma(t) \doteq \arctan_2\left(\cos(\theta(t))\dot{\phi}, \dot{\theta}\right), \tag{24}$$

where $\arctan_2$ is the four-quadrant arc-tangent function. $\gamma(t)$ is the angle between the local north, $L_N(t)$, and the wing

velocity vector, $\vec{v}(t)$, and it provides a good description of the heading of the wing while flying on the surface of the wind window.

The two lateral lines linking the wing to the GU, named steering lines, are attached to the back tips of the wing they can be pulled differentially to influence its trajectory. the prototype used for our experiments (see [46] for a detailed description), a single electric motor installed on the GU able to change the difference of length of the steering lines. In particular, a linear controller modulates the motor current in order to track a desired line length difference: such a reference is the manipulated variable u and it can be set either by a human operator through an analog joystick, or by an automatic controller.

The wing is equipped with inertial onboard sensors and a radio transmitter; the receiver and other sensors are installed on the GU, including a line angle measurement system, load cells, and an anemometer. The available sensors, together with suitable filtering algorithms, provide real-time measurements of the wing's position, velocity, and velocity angle, which can then be used for feedback control (see [47] for details on the filtering and sensor fusion aspects).

The aim of the control system is to obtain crosswind trajectories, i.e. flight paths that are symmetric and roughly perpendicular with respect to the the wind direction, whose shape is a that of an eight. This kind of patterns has been shown to be optimal for power generation, see e.g. [48], [44], [27].

The described control problem involves open-loop unstable, nonlinear and time-varying dynamics. Mathematical models for this kind of system have been derived for the design of predictive control approaches, which have been used in numerical simulations [48], [49], [44]. However, in a real-world experimental setup it is not easy to employ such approaches, due to difficulty of identifying the model parameters from experimental data, the absence of accurate measurements of the wind speed at the wing's location, finally the need to solve a nonlinear program in real-time at each time step. In a recent contribution [45], this problem has been tackled by deriving a simple model for the turning dynamics of the wing, and using a hierarchical control strategy. Here, we show that the learning approach described in Section III can be also used to solve this problem, and we compare the obtained experimental results with those achieved with the technique of [45]. Indeed a human operator is able, after some training, to obtain the desired flying paths by issuing a suitable course of the actuator position reference u, adapting to different wind conditions and counteracting the effects of turbulence and gusts: hence, we aim to design a feedback controller by learning the behavior of such a human operator from experimental data.

B. Data collection and controller learning

We have considered θ, ϕ, $\dot{\theta}$, $\dot{\phi}$, and γ as candidate feedback variables, since they can be measured or estimated with quite good accuracy and they provide a representation of the quantities that the human operator observes when controlling the wing. Moreover, they represent the position, velocity and heading of the wing, which in turn define its potential and kinetic energy. In particular, we considered four different combinations of these signals to form the feedback variable y, and we indicate the corresponding controllers as $\hat{\kappa}^i$, $i = 1, \ldots, 4$. Table I shows the choice of output vector y for these four cases. The four choices differ with respect to two aspects: (a) whether γ was included or not and (b) whether only the values at the current step were considered, or also those at a previous time step (corresponding to one second in the past). We considered aspect (a) to evaluate if γ, which in principle can be computed from the other variables (see equation (24)), is redundant. About the aspect (b), the idea was to evaluate whether the use of past measurements could yield an advantage with respect to the sensitivity to measurement noise, by implicitly providing the resulting controller with filtering capabilities.

TABLE I
CHARACTERISTICS OF THE LEARNED CONTROLLERS.

Controller	Feedback variables y	N^v	N^{nz}	T^c	$\hat{\varepsilon}$	$\hat{\gamma}_f$	$\hat{\gamma}_{g,y}$	γ^s_Δ	$\bar{\gamma}$
$\hat{\kappa}^1$	$\theta(t)$ $\theta(t-50)$ $\phi(t)$ $\phi(t-50)$ $\dot{\theta}(t)$ $\dot{\theta}(t-50)$ $\dot{\phi}(t)$ $\dot{\phi}(t-50)$ $\gamma(t)$ $\gamma(t-50)$	496	38	1025	0.003	2.04	0.38	0.038	0.31
$\hat{\kappa}^2$	$\theta(t)$ $\phi(t)$ $\dot{\theta}(t)$ $\dot{\phi}(t)$ $\gamma(t)$	136	41	458	0.0036	3.5	0.56	0.059	0.15
$\hat{\kappa}^3$	$\theta(t)$ $\phi(t)$ $\dot{\theta}(t)$ $\dot{\phi}(t)$	91	36	90	0.0042	4.2	0.62	0.11	0.11
$\hat{\kappa}^4$	$\theta(t)$ $\theta(t-50)$ $\phi(t)$ $\phi(t-50)$ $\dot{\theta}(t)$ $\dot{\theta}(t-50)$ $\dot{\phi}(t)$ $\dot{\phi}(t-50)$	325	45	505	0.0032	3.9	0.4	0.11	0.12

Legend: N^v = N. of optimization variables. N^{nz} = N. of non-zero terms. T^c = computational time. $\bar{\gamma} \doteq (1 - \hat{\gamma}_{g,y})/\hat{\gamma}_f$.

The employed sampling time is $T_s = 0.02\,\text{s}$. While the line forces and the ground wind speed and direction are measured, we do not consider such variables for feedback control, since the human operator does not exploit this information.

We have collected experimental data from a flight session of 12 minutes, i.e. about $3.5\,10^4$ data points, where the wing was controlled by a human operator. period of a single figure-eight path is about 5 s, hence a total of about 130 cycles has been used for the design of the four controllers.

We have chosen polynomial basis functions of the following form:

$$\varphi_i(y) = y_{j_1}^{\alpha} y_{j_2}^{\beta}; j_1, j_2 = 1, \ldots, n_y;\ \alpha, \beta = 0, \ldots, 3. \quad (25)$$

For each combination of $j_1\,, j_2,\, \alpha\,, \beta$ in (25), we assigned a progressive index $i = 1, \ldots, M$ to the corresponding polynomial, where $M = (q\,n_y + 1)(q\,n_y + 2)/2$. Note that using orthogonal polynomials leads to a combinatorial number of basis functions. For this reason, a different choice has been made here: the basis functions have been generated as products of univariate polynomials. This is a simple and effective technique that allows us to avoid the combinatorial explosion and to have at the same time a set of functions giving accurate approximations. Note also that, using ad-hoc algorithms for solving the required optimization problems, it is nowadays possible to deal with hundred thousands basis functions. Even with such large numbers of functions, the

ℓ_1 norm minimization may lead to simple representations, eliminating all those terms that are not needed.

For all four feedback variable combinations, we run Algorithm (1) to obtain four controllers. The value $\alpha = 1.1$ has been used in all the four cases. Other values of α have also been considered: results similar to those obtained with $\alpha = 1.1$ have been observed for $1.1 \lesssim \alpha \lesssim 1.6$. Values $\alpha \gtrsim 1.6$ led to a degradation of the controller approximation accuracy. Table I shows, for each designed controller, the number of optimization variables, the resulting non-zero components, the times required to derive the controller, and the obtained values of the design parameters involved in Algorithm (1). The following observations can be made about this table:

1) Starting from a quite large number of decision variables, the approach yields control laws with a few non-zero terms, which we implemented on a real-time machine using the xPC Target® toolbox of Matlab®.
2) The computational times for the learning phase (referred to a laptop with 2.8 Ghz core i7 processor, 8 GB RAM and the CVX tool [50]) resulted to be quite low. Indeed, a quite large number of data have been used for design: $N = 10501$. The data with indexes $k = -N, \ldots, -60, -30, -1$ have been used to form the constraints (b) in (18) and in (19) (we did not use all the data due to memory saturation). The total number of constraints in (18)-(b) and (19)-(b) resulted to be 61075.
3) The times required for the on-line control computation were of the order of 10^{-5} s (including the time required to condition the measured signals and to log the test results on a hard-drive), far below the employed sampling time of 0.02 s.
4) The controllers $\hat{\kappa}^1$ and $\hat{\kappa}^2$ satisfy the (estimated) stability condition $\gamma^s_\Delta < \bar{\gamma}$ with a "quite large" gap, while $\hat{\kappa}^3$ satisfies it with a smaller gap and $\hat{\kappa}^3$ with a very small gap, so that $\gamma^s_\Delta \simeq \bar{\gamma}$ (in the table the values are equal due to rounding). As we point out in the next Section IV-C, the controllers 1, 2 and 4 were able to stabilize the closed-loop system, with controller 4 having significantly worse performance than 1 and 2, while $\hat{\kappa}^3$ was not able to stabilize the system, causing several crashes.

Remark 6: As discussed in the previous sections, one important feature of our approach is that, if an adequate number of data is used and the conditions of the stability theorem are satisfied, we can be confident that the controller will work correctly. Moreover, the observation 4) above evidences another important aspect, i.e. that the approach allows one to obtain effective indications on the stabilizing properties of the designed controllers: if $\gamma^s_\Delta < \bar{\gamma}$ with a "large" gap, then we can be confident that the controller will work correctly. If $\gamma^s_\Delta \simeq \bar{\gamma}$, then the controller will likely not work correctly, leading to instability or low robustness. □

C. Experimental results and discussion

We tested the controllers in 15-minutes-long experiment batches (including take-off and landing phases), each one corresponding to roughly 140 full cycles of autonomous flight.

TABLE II
EXPERIMENTAL RESULTS OF 15-MINUTES TEST BATCHES: AVERAGE POSITION AND RELATED DEVIATIONS (RAD), AND AVERAGE WIND SPEED AT 4 M ABOVE THE GROUND (M/S).

	$\overline{\theta}$	$\overline{\phi}$	$\Delta\theta$	$\Delta\phi$	$\overline{\Delta\theta}$	$\overline{\Delta\phi}$	$\overline{W}$
$\hat{\kappa}^1$	0.61	0.02	0.03	0.03	0.09	0.1	2.9
$\hat{\kappa}^2$	0.62	0.01	0.06	0.02	0.13	0.06	3.2
$\hat{\kappa}^3$	not stable						2.9
$\hat{\kappa}^4$	0.54	-0.02	0.14	0.11	0.25	0.17	3.5
κ	0.79	0.02	0.08	0.08	0.33	0.35	5.5
κ^0	0.53	0.01	0.04	0.03	0.09	0.08	2.5

We denote with $\overline{\theta}_k$, $\overline{\phi}_k$ the average position, in spherical coordinates, of the kth full cycle, and with K the total number of cycles carried out in a test. In order to evaluate the results obtained with the different controllers and to compare them with the data related to manual flight and to the automatic control approach described in [45], we computed the following quantities:

$$\begin{aligned} \overline{\phi} &\doteq \tfrac{1}{K}\sum_{k=1}^{K}\overline{\phi}_k, & \overline{\theta} &\doteq \tfrac{1}{K}\sum_{k=1}^{K}\overline{\theta}_k \\ \Delta\phi &\doteq \sqrt{\tfrac{1}{K-1}\sum_{k=1}^{K}(\overline{\phi}_k-\overline{\phi})^2}, & \Delta\theta &\doteq \sqrt{\tfrac{1}{K-1}\sum_{k=1}^{K}(\overline{\theta}_k-\overline{\theta})^2} \\ \overline{\Delta\phi} &\doteq \max_{k=1,\ldots,K}|\overline{\phi}_k-\overline{\phi}|, & \overline{\Delta\theta} &\doteq \max_{k=1,\ldots,K}|\overline{\theta}_k-\overline{\theta}|. \end{aligned} \tag{26}$$

The variables (26) provide the average position of the trajectories flown during each test, as well as an indication on the average and maximal deviation of each single flown trajectory from the overall average. The choice of these quantities as performance indicators is motivated by the fact that theoretical, numerical and experimental results show that, within quite slack limits, the most important aspect for the sake of power generation is the average position of a flown path, rather than its shape [51]. Hence, the flight control system shall achieve flying paths with consistent average position, and a flight controller can be considered to be "better" than another if it is able to obtain trajectories whose average position is less variable, i.e. with smaller values of $\Delta\phi$, $\Delta\theta$, $\overline{\Delta\phi}$, $\overline{\Delta\phi}$. The task of regulating the power output, i.e. of setting the average flown position according to the desired corresponding average power, can then be carried out by a supervisory control approach, like the adaptive strategy proposed in [51].

The results obtained with the four designed controllers are reported in Table II. The table also shows the performance achieved by the human operator, indicated as κ, in the test whose data have been used to learn the controllers, and by a controller designed with the approach described in [45], indicated as κ^0. The learned controllers $\hat{\kappa}^1$, $\hat{\kappa}^2$ were able to keep the wing's path inside the wind window and to stabilize the system according to Definition 1, achieving a good consistency of average position of the flown paths. The differences in average position achieved by these controllers with respect to κ and κ^0 are due to the difference in wind speed during the tests, whose average value $\overline{W}$ measured at 4 m above the ground is shown in Table II. wind speed can be considered as an exogenous, unmeasured disturbance, which can be embedded in the variable $e_s(t)$ of (2). Different wind

speeds induce a change in the position of the closed loop trajectories in the ϕ, θ plane. In particular, the lower was the wind speed, the closer were the flight paths to the ground. This result is consistent both with the theoretical results of Section III-A and with physical considerations on the system. The controller $\hat{\kappa}^4$ was able to achieve figure-eight trajectories, however with quite poor repeatability as evidenced by the high average and maximal deviations. Finally, the controller $\hat{\kappa}^3$ was not able to keep the wing airborne and the closed loop trajectories gradually neared the ground until the wing crashed. Considering that the same data set and the same form of the basis functions were used to design all the controllers, the main reason for such differences in performance lie in the choice of the feedback variables. The ones used by $\hat{\kappa}^3$, namely the current position and velocity of the wing, were not sufficient to extrapolate with high enough accuracy the behavior of the human operator. The use of the same values at the current time and 1 second in the past, adopted by $\hat{\kappa}^4$, yielded a controller able to stabilize the plant, but whose performance in terms of variability of the flown paths were the worst. The use of the current position plus the velocity angle in controller $\hat{\kappa}^2$ provided much better results (a movie of the experimental tests with the controllers $\hat{\kappa}^2$ and $\hat{\kappa}^4$ is available online [28]). It has to be noted that the velocity angle is determined by the wing's position and velocity, so in principle the feedback variables used by controller $\hat{\kappa}^3$ provide the same information as those used by $\hat{\kappa}^2$. However such a relationship, given by (24), is not described exactly by the polynomials (25), hence the learning algorithm was not able to extract this information. The controller $\hat{\kappa}^1$, which uses the same feedback variables as $\hat{\kappa}^2$ both at the current time and with 1 second delay, gives performance similar to $\hat{\kappa}^2$, hence indicating that these additional variables do not provide significant new information. We remark that the controllers $\hat{\kappa}^1$ and $\hat{\kappa}^2$ have similar complexity in terms of number of non-zero terms, despite the fact that $\hat{\kappa}^1$ uses twice as many variables as $\hat{\kappa}^2$: this effect is due to the use of the ℓ_1-norm cost function in (18), which encourages sparsity and hence the automatic selection of the terms (hence also of the feedback variables) whose importance is higher.

In summary, the obtained results highlight that 1) with the proposed approach, the control design effort lies mainly in the choice of the feedback variables and of the basis functions, 2) physical insight still plays an important role in selecting the most appropriate variables (like the velocity angle γ in this specific case), 3) the proposed learning algorithm is able to provide not only an approximated controller, but also indications on its stabilizing properties and performance (as commented at the end of Section IV-B).

Finally, we comment on the comparison between the performance obtained by $\hat{\kappa}^1$ and $\hat{\kappa}^2$, which gave the best results among the learned controllers, and those of the human operator κ and of the model-based controller κ^0. From Table II, it can be noted that the learned controllers were able to obtain less erratic flight trajectories than those pertaining to the identification data collected during manual flight, which are affected by larger deviations in the average trajectory position. The relatively poor performance of the human operator can be due to inexperience but also fatigue and loss of concentration. These aspects can be regarded to as disturbances acting on the input variable, and their effect is accounted for by the design parameter $\hat{\varepsilon}$. The experimental results indicate that the proposed approach is able to cope effectively with outliers and dispersed data sets caused by such disturbances.

The controller κ^0, which is derived with a model-based approach [45], achieved performance similar to those of the learned controllers (see Table II). A comparison between a complete flown loop obtained with controllers κ^0 and $\hat{\kappa}^2$ is reported in Fig. 4. It can be noted that the obtained input and output trajectories are very similar. Overall, the learned controllers and the model-based one achieved comparable results, with the notable difference that $\hat{\kappa}^2$ was designed without any knowledge on the system dynamics, while the design of κ^0 required a significant amount of prior knowledge and modeling efforts.

V. Conclusions

We presented an approach to learn a nonlinear control law directly from the data measured on an existing feedback control system, where both the controller and the plant are not known. The approach has been described by addressing and connecting theoretical stability aspects, computational algorithms and real-world implementation. A theoretical analysis unveiled that closed loop stability depends on the variability of the approximation error function, expressed in terms of its Lipschitz constant, over the set of feedback variables. This finding led to the inclusion of a new constraint in the regularized optimization approach employed to learn the controller. The learning algorithm involves the solution of convex optimization problems only, and it is shown to provide a stabilizing controller as **the number of employed data is sufficiently large (but finite)**. We finally presented the experimental application of the approach to the problem of controlling the crosswind flight of tethered wings, which involves nonlinear, open-loop unstable dynamics subject to unmeasured external disturbances. In this application, the learning technique has been successfully employed to learn the behavior of a human controller and derive an automatic control law without any explicit knowledge of the system's dynamics.

The next step along this line of research is to extend the approach to on-line learning and improvement of the controller, using the measured input-output data, while still retaining stability guarantees.

Appendix

Derivation of equation (8). Consider a generic time instant t. Using property (5), we have:

$$\begin{aligned} \|y(t+1)\| &= \|g(y(t), e_s(t))\| \\ &\leq \|g(y(t), 0)\| + \gamma_{g,e}\|e_s(t)\|. \end{aligned}$$

Using property (7), we obtain

$$\|y(t+1)\| \leq \gamma_{g,y}\|y(t)\| + \gamma_{g,e}\|e_s(t)\|.$$

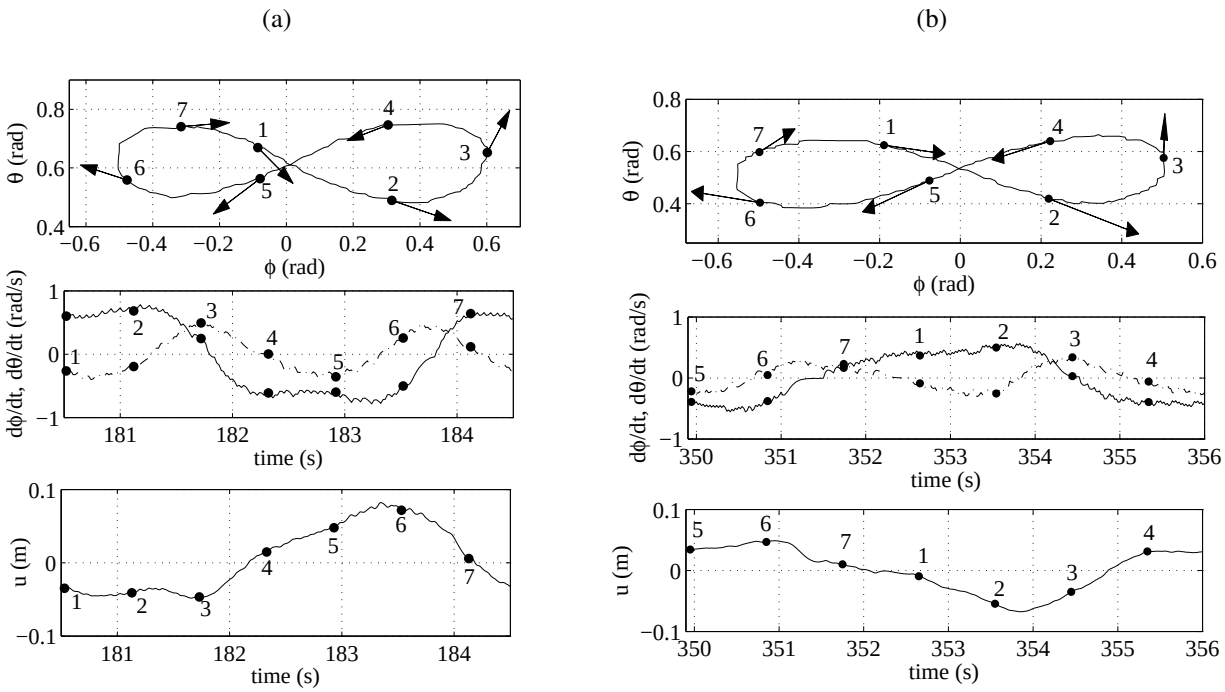

Fig. 4. Experimental results for (a) the learned controller $\hat{\kappa}^2$ and (b) the model-based controller κ^0. From top: wing's path in ϕ, θ coordinates and related gradient estimated by the velocity angle γ (arrows); angular velocities $\dot{\phi}$ (solid) and $\dot{\theta}$ (dash-dot); control input u.

The result is then established by applying recursively this inequality from $t = 0$ and using the definition of the ℓ_∞ signal norm $\|\mathbf{e}_s\|_\infty$:

$$\begin{aligned}\|y(t)\| &\leq \gamma_{g,y}^t\|y(0)\| + \sum_{k=0}^{t-1}\left(\gamma_{g,y}^k\gamma_{g,e}\|\mathbf{e}_s\|_\infty\right)\\ &\leq \frac{1}{1-\gamma_{g,y}}\gamma_{g,e}(\|\mathbf{e}_s\|_\infty) + \gamma_{g,y}^t\|y(0)\|,\ \forall t \geq 0.\end{aligned}$$

Proof of Theorem 1. Consider $t = 0$. We have

$$\begin{aligned}&\|y(1)\| = \|\hat{g}(y(0), e_s(0), e_y(0))\|\\ &= \|\hat{g}(y(0), e_s(0), e_y(0)) - g(y(0), e_s(0)) + g(y(0), e_s(0))\|.\end{aligned}$$

Using properties (3), (5) and (7),

$$\begin{aligned}&\|\hat{g}(y(0), e_s(0), e_y(0)) - g(y(0), e_s(0)) + g(y(0), e_s(0))\|\\ &\leq \|\hat{g}(y(0), e_s(0), e_y(0)) - g(y(0), e_s(0))\| + \|g(y(0), e_s(0))\|\\ &= \|f(y(0), \hat{\kappa}(y(0) + e_y(0)), e_s(0)) - f(y(0), \kappa(y(0)), e_s(0))\|\\ &+\|g(y(0), e_s(0))\|\\ &\leq \gamma_f|\hat{\kappa}(y(0) + e_y(0)) - \kappa(y(0))| + \gamma_{g,y}\|y(0)\| + \gamma_{g,e}\|e_s(0)\|\end{aligned}$$

Using properties (11) and (13),

$$\begin{aligned}&\gamma_f|\hat{\kappa}(y(0) + e_y(0)) - \kappa(y(0))| + \gamma_{g,y}\|y(0)\| + \gamma_{g,e}\|e_s(0)\|\\ &= \gamma_f|\hat{\kappa}(y(0) + e_y(0))) - \hat{\kappa}(y(0)) + \hat{\kappa}(y(0)) - \kappa(y(0))|\\ &+\gamma_{g,y}\|y(0)\| + \gamma_{g,e}\|e_s(0)\|\\ &= \gamma_f|\Delta(y(0)) - \Delta(0) + \Delta_0| + \gamma_f\gamma_{\hat{\kappa}}\|e_y(0)\|\\ &+\gamma_{g,y}\|y(0)\| + \gamma_{g,e}\|e_s(0)\|\\ &\leq \gamma_f\gamma_\Delta\|y(0)\| + \gamma_f|\Delta_0| + \gamma_f\gamma_{\hat{\kappa}}\|e_y(0)\|\\ &+\gamma_{g,y}\|y(0)\| + \gamma_{g,e}\|\mathbf{e}_s\|\\ &\leq (\gamma_f\gamma_\Delta + \gamma_{g,y})\|y(0)\| + \gamma_{g,e}\|\mathbf{e}_s\|_\infty + \gamma_f|\Delta_0| + \gamma_f\gamma_{\hat{\kappa}}\varepsilon_y\\ &= \gamma\|y(0)\| + \gamma_{g,e}\|\mathbf{e}_s\|_\infty + \gamma_f|\Delta_0| + \gamma_f\gamma_{\hat{\kappa}}\varepsilon_y\end{aligned}$$

where we recall that $\Delta(0,0) \doteq \Delta_0$ and $\gamma \doteq (\gamma_f\gamma_\Delta + \gamma_{g,y}) < 1$. Analogously, for $t = 1$, we have that

$$\begin{aligned}&\|y(2)\| \leq \gamma\|y(1)\| + \gamma_{g,e}\|\mathbf{e}_s\|_\infty + \gamma_f|\Delta_0| + \gamma_f\gamma_{\hat{\kappa}}\varepsilon_y\\ &\leq \gamma^2\|y(0)\| + \gamma\gamma_{g,e}\|\mathbf{e}_s\|_\infty + \gamma\left(\gamma_f|\Delta_0| + \gamma_f\gamma_{\hat{\kappa}}\varepsilon_y\right)\\ &+\gamma_{g,e}\|\mathbf{e}_s\|_\infty + \gamma_f|\Delta_0| + \gamma_f\gamma_{\hat{\kappa}}\varepsilon_y.\end{aligned}$$

Generalizing to any $t \geq 0$, we obtain

$$\begin{aligned}\|y(t)\| \leq \gamma^t\|y(0)\| + \sum_{k=0}^{t-1}\gamma^k\gamma_{g,e}\|\mathbf{e}_s\|_\infty\\ + \sum_{k=0}^{t-1}\left(\gamma^k\left(\gamma_f|\Delta_0| + \gamma_f\gamma_{\hat{\kappa}}\varepsilon_y\right)\right).\end{aligned}$$

Considering (15) and the convergence of the geometric series, it follows that

$$\begin{aligned}\|y(t)\| \leq \frac{\gamma_{g,e}}{1-\gamma}(\|\mathbf{e}_s\|_\infty) + \gamma^t\|y(0)\|\\ +\frac{1}{1-\gamma}\left(\gamma_f|\Delta_0| + \gamma_f\gamma_{\hat{\kappa}}\varepsilon_y\right),\ \forall t \geq 0,\end{aligned}$$

which proves the claim.

Proof of Theorem 2. Consider step 2) of Algorithm 2. Equations (20) imply that

$$\begin{aligned}&\delta\tilde{z}_t = \max_{i,j\in J_t}|\widetilde{z}(i) - \widetilde{z}(j)|\\ &= \max_{i,j\in J_t}|\mathfrak{f}(\widetilde{w}(i)) - \mathfrak{f}(\widetilde{w}(j)) + e(i) - e(j)|\\ &\geq \max_{i,j\in J_t}\left(|e(i) - e(j)| - |\mathfrak{f}(\widetilde{w}(i)) - \mathfrak{f}(\widetilde{w}(j))|\right).\end{aligned}$$

From Assumption 5, it follows that, for any $\lambda > 0$, there exist a sufficiently large N and two pairs $(\widetilde{w}(i), e(i)) \in \{\widetilde{w}(t)\}_{t=-N}^{-1} \times B_e$ and $(\widetilde{w}(j), e(j)) \in \{\widetilde{w}(t)\}_{t=-N}^{-1} \times B_e$ with $i, j \in J_t$ such that

$$|\varepsilon - e(i)| \leq \lambda,\ |-\varepsilon - e(j)| \leq \lambda,$$

thus yielding the following inequality:

$$|e(i) - e(j)| \geq 2\varepsilon - 2\lambda.$$

Moreover, due the Lipschitz continuity property, we have

$$|\mathfrak{f}(\widetilde{w}(i)) - \mathfrak{f}(\widetilde{w}(j))| \leq \gamma_{\mathfrak{f}}\|\widetilde{w}(i) - \widetilde{w}(j)\| \leq 2\gamma_{\mathfrak{f}}\rho.$$

The above inequalities imply that

$$\begin{aligned}&\delta\tilde{z}_t \geq \max_{i,j\in J_t}\left(|e(i) - e(j)| - |\mathfrak{f}(\widetilde{w}(i)) - \mathfrak{f}(\widetilde{w}(j))|\right)\\ &\geq \max_{i,j\in J_t}\left(|e(i) - e(j)| - 2\gamma_{\mathfrak{f}}\rho\right)\\ &\geq 2\varepsilon - 2\lambda - 2\gamma_{\mathfrak{f}}\rho.\end{aligned} \tag{27}$$

On the other hand,

$$\begin{aligned}\delta\tilde{z}_t &= \max_{i,j\in J_t} |\widetilde{z}(i)-\widetilde{z}(j)| \\ &= \max_{i,j\in J_t} |\mathfrak{f}(\widetilde{w}(i))-\mathfrak{f}(\widetilde{w}(j))+e(i)-e(j)| \\ &\le \max_{i,j\in J_t} (|e(i)-e(j)|+|\mathfrak{f}(\widetilde{w}(i))-\mathfrak{f}(\widetilde{w}(j))|) \\ &\le 2\varepsilon+2\gamma_{\mathfrak{f}}\rho.\end{aligned} \tag{28}$$

Since λ and ρ can be chosen arbitrarily small, from (27) and (28) it follows that $\delta\tilde{z}_t \to 2\varepsilon$ as $N\to\infty$, i.e. that $\delta\tilde{z}_t/2 \to \varepsilon$. In step 3) of Algorithm 2, the operation of taking the mean over all $\delta\tilde{z}_t/2$ is inessential in this analysis. It can be effective in the finite data case in order to not under-estimate or over-estimate ε.

Proof of Theorem 3. Define

$$(\bar{w}^1,\bar{w}^2) \doteq \arg\max_{w^1,w^2\in W} \frac{\left|\mathfrak{f}(w^1)-\mathfrak{f}(w^2)\right|}{\|w^1-w^2\|}$$

and, without loss of generality, suppose that $\mathfrak{f}(\bar{w}^1) > \mathfrak{f}(\bar{w}^2)$. From Assumption 5, it follows that, for any $\lambda>0$, there exist a sufficiently large N and two pairs $(\widetilde{w}(i),e(i)) \in \{\widetilde{w}(t)\}_{t=-N}^{-1}\times B_e$ and $(\widetilde{w}(j),e(j)) \in \{\widetilde{w}(t)\}_{t=-N}^{-1}\times B_e$ with $i,j\in\{-N,\dots-1\}$ such that

$$\begin{gathered}\left\|\bar{w}^1-\widetilde{w}(i)\right\|\le\lambda,\ \left\|\bar{w}^2-\widetilde{w}(j)\right\|\le\lambda \\ |\varepsilon-e(i)|\le\lambda,\ |-\varepsilon-e(j)|\le\lambda.\end{gathered} \tag{29}$$

Moreover,

$$\begin{aligned}\left\|\bar{w}^1-\bar{w}^2\right\| &= \left\|\bar{w}^1-\widetilde{w}(i)-\bar{w}^2+\widetilde{w}(j)+\widetilde{w}(i)-\widetilde{w}(j)\right\| \\ &\ge \|\widetilde{w}(i)-\widetilde{w}(j)\|-\left\|\bar{w}^1-\widetilde{w}(i)\right\|-\left\|\bar{w}^2-\widetilde{w}(j)\right\| \\ &\ge \|\widetilde{w}(i)-\widetilde{w}(j)\|-2\lambda.\end{aligned} \tag{30}$$

Also, from the Lipschitz continuity property of $\mathfrak{f}$ and from (20), we have that

$$\begin{aligned}\mathfrak{f}(\bar{w}^1)-\mathfrak{f}(\bar{w}^2) &\le \mathfrak{f}(\widetilde{w}(i))-\mathfrak{f}(\widetilde{w}(j))+2\gamma_{\mathfrak{f}}\lambda \\ &= \widetilde{z}(i)-e(i)-\widetilde{z}(j)+e(j)+2\gamma_{\mathfrak{f}}\lambda \\ &\le \widetilde{z}(i)-\widetilde{z}(j)-2\varepsilon+2\lambda+2\gamma_{\mathfrak{f}}\lambda\end{aligned} \tag{31}$$

where the last inequality follows from (29). Considering that $\mathfrak{f}(\bar{w}^1) > \mathfrak{f}(\bar{w}^2)$, inequalities (30) and (31) imply that

$$\frac{\left|\mathfrak{f}(\bar{w}^1)-\mathfrak{f}(\bar{w}^2)\right|}{\|\bar{w}^1-\bar{w}^2\|} = \frac{\mathfrak{f}(\bar{w}^1)-\mathfrak{f}(\bar{w}^2)}{\|\bar{w}^1-\bar{w}^2\|} \le \frac{\widetilde{z}(i)-\widetilde{z}(j)-2\varepsilon+2\lambda+2\gamma_{\mathfrak{f}}\lambda}{\|\widetilde{w}(i)-\widetilde{w}(j)\|-2\lambda}.$$

Since λ is arbitrarily small, we have that, as $N\to\infty$,

$$\frac{\left|\mathfrak{f}(\bar{w}^1)-\mathfrak{f}(\bar{w}^2)\right|}{\|\bar{w}^1-\bar{w}^2\|} \le \frac{\widetilde{z}(i)-\widetilde{z}(j)-2\varepsilon}{\|\widetilde{w}(i)-\widetilde{w}(j)\|}.$$

But $\frac{\left|\mathfrak{f}(\bar{w}^1)-\mathfrak{f}(\bar{w}^2)\right|}{\left\|\bar{w}^1-\bar{w}^2\right\|} = \gamma_{\mathfrak{f}}$ and $\frac{\widetilde{z}(i)-\widetilde{z}(j)-2\varepsilon}{\|\widetilde{w}(i)-\widetilde{w}(j)\|} \le \hat{\gamma}$, where it has been considered that, from Theorem 2, $\lim_{N\to\infty}\hat{\varepsilon}=\varepsilon$. It follows that, as $N\to\infty$,

$$\gamma_{\mathfrak{f}} \le \hat{\gamma}. \tag{32}$$

On the other hand, since $|e(t)|\le\varepsilon$, $\forall t$,

$$\begin{aligned}\tilde{\gamma}_{ij} = \frac{\widetilde{z}(i)-\widetilde{z}(j)-2\varepsilon}{\|\widetilde{w}(i)-\widetilde{w}(j)\|} &\le \frac{\mathfrak{f}(\widetilde{w}(i))-\mathfrak{f}(\widetilde{w}(j))+e(i)-e(j)-2\varepsilon}{\|\widetilde{w}(i)-\widetilde{w}(j)\|} \\ &\le \frac{\mathfrak{f}(\widetilde{w}(i))-\mathfrak{f}(\widetilde{w}(j))}{\|\widetilde{w}(i)-\widetilde{w}(j)\|} \le \gamma_{\mathfrak{f}},\ \forall i,j.\end{aligned}$$

It follows that $\gamma_{\mathfrak{f}}\ge\hat{\gamma}$, which, together with (32), proves the claim.

Proof of Theorem 4. **Consider $\varepsilon_s', \varepsilon_s'' > 0$ such that $\varepsilon_s'+\varepsilon_s''=\varepsilon_s$. Let δ_f and $\delta_{g,y}$ be two positive constants such that**

$$\begin{aligned}\delta_f &< \frac{\varepsilon_s'\gamma_f^2}{2+2\gamma_{g,y}+\varepsilon_s'\gamma_f} \\ \delta_{g,y} &< \frac{\varepsilon_s'\left(\gamma_f-\delta_f\right)}{2}.\end{aligned}$$

According to Theorem 3, two finite integers $N_f, N_{g,y}$ exist such that

$$\begin{aligned}|\gamma_f-\hat{\gamma}_f| &\le \delta_f, \qquad \forall N>N_f \\ |\gamma_{g,y}-\hat{\gamma}_{g,y}| &\le \delta_{g,y}, \quad \forall N>N_{g,y}.\end{aligned}$$

Then, through straightforward passages,

$$\begin{aligned}\left|\frac{1-\gamma_{g,y}}{\gamma_f}-\bar{\gamma}\right| &= \left|\frac{1-\gamma_{g,y}}{\gamma_f}-\frac{1-\hat{\gamma}_{g,y}}{\hat{\gamma}_f}\right| \\ &\le \frac{\delta_f\left(1+\gamma_{g,y}\right)}{\gamma_f^2-\delta_f\gamma_f}+\frac{\gamma_f\delta_{g,y}}{\gamma_f^2-\delta_f\gamma_f} < \varepsilon_s'.\end{aligned}$$

Since $\gamma_\Delta' = \bar{\gamma}-\varepsilon_s$ (see step 3 of Algorithm 1), we have that

$$\gamma_\Delta' < \frac{1-\gamma_{g,y}}{\gamma_f}+\varepsilon_s'-\varepsilon_s = \frac{1-\gamma_{g,y}}{\gamma_f}-\varepsilon_s''$$

which implies that

$$\gamma^s{}_\Delta < \frac{1-\gamma_{g,y}}{\gamma_f}-\varepsilon_s'', \tag{33}$$

being $\gamma^s{}_\Delta \le \gamma_\Delta'$ (see step 5 of Algorithm 1).

Following the same lines of the proof of Theorem 3 in [16], it can be shown that

$$\limsup_{N\to\infty}\gamma_\Delta \le \gamma_\Delta^s.$$

That is, there exists a finite integer N_{γ_Δ} such that

$$\gamma_\Delta \le \gamma_\Delta^s+\varepsilon_s''. \tag{34}$$

Taking $N=\max\left(N_f,N_{g,y},N_{\gamma_\Delta}\right)<\infty$, from (33) and (34) it follows that

$$\gamma_\Delta < \frac{1-\gamma_{g,y}}{\gamma_f},$$

hence satisfying the stability condition of Theorem 1 and proving the claim.

REFERENCES

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.

[2] S. Battilotti, "Robust stabilization of nonlinear systems with pointwise norm-bounded uncertainties: a control lyapunov function approach," *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 3 – 17, 1999.

[3] S. Nersesov and M. Haddad, "On the stability and control of nonlinear dynamical systems via vector lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 51, no. 2, pp. 203 – 215, 2006.

[4] L. Freidovich and H. Khalil, "Performance recovery of feedback-linearization-based designs," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2324 – 2334, 2008.

[5] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transaction on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[6] F.-C. Chen and H. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Transactions on Automatic Control*, vol. 40, no. 5, pp. 791–801, 1995.

[7] K. J. Astrom and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1995.

[8] A. Levin and K. Narendra, "Control of nonlinear dynamical systems using neural networks. II. observability, identification, and control," *IEEE Transaction on Neural Networks*, vol. 7, no. 1, pp. 30–42, 1996.

[9] M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 447–451, 1996.

[10] J. Cabrera and K. S. Narendra, "Issues in the application of neural networks for tracking based on inverse control," *IEEE Transaction on Automatic Control*, vol. 44, no. 11, pp. 2007–2027, 1999.

[11] A. Yesildirek and L. Lewis, "Feedback linearization using neural networks," *Automatica*, vol. 31, no. 11, pp. 1659–1664, 1995.

[12] J. Spall and A. Cristion, "Model-free control of nonlinear stochastic systems with discrete-time measurements," *IEEE Transactions on Automatic Control*, vol. 43, no. 9, pp. 1198–1210, 1998.

[13] G. C. M. D. Abreu, R. L. Teixeira, and J. F. Ribeiro, "A neural network-based direct inverse control for active control of vibrations of mechanical systems," in *Proceedings of the sixth Brazilian Symposium on Neural Networks*, 2000, pp. 107–112.

[14] M. Campi and S. Savaresi, "Direct nonlinear control design: The virtual reference feedback tuning (VRFT) approach," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 14–27, 2006.

[15] D. B. Anuradha, G. P. Reddy, and J. S. N. Murthy, "Direct inverse neural network control of a continuous stirred tank reactor (cstr)," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, 2009.

[16] C. Novara, L. Fagiano, and M. Milanese, "Direct feedback control design for nonlinear systems," *Automatica*, vol. 49, no. 4, pp. 849–860, 2013.

[17] S. Formentin, P. De Filippi, M. Corno, M. Tanelli, and S. Savaresi, "Data-driven design of braking control systems," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 186–193, 2013.

[18] K. J. Hunt and D. Sbarbaro, "Neural networks for nonlinear internal model control," *Control Theory and Applications, IEE Proceedings D*, vol. 138, no. 5, pp. 431–438, 1991.

[19] M. Brown, G. Lightbody, and G. Irwin, "Nonlinear internal model control using local model networks," *IEE Proceedings - Control Theory and Applications*, vol. 144, no. 6, pp. 505–514, 1997.

[20] F. Lewis, D. Vrabie, and K. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *Control Systems, IEEE*, vol. 32, no. 6, pp. 76–105, 2012.

[21] S. Yin, H. Luo, and S. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2402–2411, 2014.

[22] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567 – 1581, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959152412001503

[23] M. Norgaard, O. Ravn, N. Poulsen, and L. K. Hansen, *Neural Networks for Modeling and Control of Dynamic Systems*. Springer, 2000.

[24] T. Urbančič and I. Bratko, *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994, ch. 13. Learning to Control Dynamic Systems, pp. 246–261, freely available: http://www1.maths.leeds.ac.uk/∼charles/statlog/.

[25] M. Khammash and H. El-Samad, "Systems biology: from physiology to gene regulation," *Control Systems Magazine, IEEE*, vol. 24, no. 4, pp. 62–76, 2004.

[26] L. Fagiano, M. Milanese, and D. Piga, "High-altitude wind power generation," *IEEE Transactions on Energy Conversion*, vol. 25, no. 1, pp. 168 –180, mar. 2010.

[27] L. Fagiano and M. Milanese, "Airborne wind energy: an overview," in *American Control Conference 2012*, Montreal, Canada, 2012, pp. 3132–3143.

[28] Lerning a nonlinear controller from data, experimental test movie, October 2012. Available on–line: https://fagiano.faculty.polimi.it/movies/LC_kite_LD.mp4.

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

[30] E. Bai, H. Cho, and R. Tempo, "Convergence properties of the membership set," *Automatica*, vol. 34, no. 10, pp. 1245–1249, 1998.

[31] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231 – 2242, oct. 2004.

[32] J. Fuchs, "Recovery of exact sparse representations in the presence of bounded noise," *Information Theory, IEEE Transactions on*, vol. 51, no. 10, pp. 3601 –3608, oct. 2005.

[33] J. Tropp, "Just relax: convex programming methods for identifying sparse signals in noise," *Information Theory, IEEE Transactions on*, vol. 52, no. 3, pp. 1030 –1051, mar. 2006.

[34] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *Information Theory, IEEE Transactions on*, vol. 52, no. 1, pp. 6 – 18, jan. 2006.

[35] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489 – 509, feb. 2006.

[36] C. Novara, "Sparse identification of nonlinear functions and parametric set membership optimality analysis," *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 3236–3241, 2012.

[37] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B.Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, pp. 1691–1723, 1995.

[38] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[39] F. Girosi., "An equivalence between sparse approximation and support vector machines," *Neural Computation*, vol. 10, no. 6, pp. 1455–1480, 1998.

[40] K. Hsu, C. Novara, T. Vincent, M. Milanese, and K. Poolla, "Parametric and nonparametric curve fitting," *Automatica*, vol. 42/11, pp. 1869–1873, 2006.

[41] C. Novara, T. Vincent, K. Hsu, M. Milanese, and K. Poolla, "Parametric identification of structured nonlinear systems," *Automatica*, vol. 47, no. 4, pp. 711 – 721, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109811000781

[42] H. White, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, pp. 246–257, 1991.

[43] M. Canale, L. Fagiano, and M. Milanese, "Power kites for wind energy generation," *IEEE Control Systems Magazine*, vol. 27, no. 6, pp. 25–38, December 2007.

[44] ——, "High altitude wind energy generation using controlled power kites," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 279 –293, mar. 2010.

[45] L. Fagiano, A. Zgraggen, M. Morari, and M. Khammash, "Automatic crosswind flight of tethered wings for airborne wind energy: modeling, control design and experimental results," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1433–1447, 2014.

[46] L. Fagiano and T. Marks, "Design of a small-scale prototype for research in airborne wind energy," *IEEE/ASME Transactions on Mechatronics*, to appear, doi: 10.1109/TMECH.2014.2322761 2014.

[47] L. Fagiano, K. Huynh, B. Bamieh, and M. Khammash, "On sensor fusion for airborne wind energy systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 930–943, 2014.

[48] B. Houska and M. Diehl, "Optimal control for power generating kites," in 9^{th} *European Control Conference*, Kos, GR, 2007, pp. 3560–3567.

[49] P. Williams, B. Lansdorp, and W. Ockels, "Optimal crosswind towing and power generation with tethered kites," *Journal of guidance, control, and dynamics*, vol. 31, pp. 81–93, 2008.

[50] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Aug. 2010.

[51] A. Zgraggen, L. Fagiano, and M. Morari, "Real-time optimization and adaptation of the crosswind flight of tethered wings for airborne wind energy," *IEEE Transactions on Control Systems Technology*, to appear, 2014.