Automatic crosswind flight of tethered wings for airborne wind energy: a direct data-driven approach ⋆

Lorenzo Fagiano * and **Carlo Novara** **

* *ABB Schweiz Ltd., Corporate Research Center, Baden-Daettwil - Switzerland; (e-mail: lorenzo.fagiano@ch.abb.com).*
** *Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino - Italy (e-mail: carlo.novara@polito.it)*

Abstract: A control design approach for the autonomous crosswind flight of tethered wings, to be used in airborne wind energy systems, is described. The proposed technique aims to learn the behavior of a human pilot, by exploiting a finite number of data collected during manual operation of the system, hence avoiding the need for an explicit model of the system dynamics. Along with the design technique, a theoretical analysis as well as experimental data showing its successful application are presented.

1. INTRODUCTION

Airborne wind energy technologies aim to produce power exploiting the aerodynamic forces generated by a wing tethered to the ground, hence realizing a wind generator capable of reaching higher altitudes with respect to the current wind turbines, see e.g. Ahrens et al. [2013], Fagiano and Milanese [2012] for details. In particular, the lift force developed by the wing is large enough to sustain its weight and that of the tethers, and to produce power. In the configuration considered in this work, such a power is extracted at ground level by executing a two-phase ("pumping") cycle consisting of reeling-out the lines under large traction forces and subsequently reeling-in under low forces, thus realizing a positive net energy balance ~~Ahrens et al. [2013]~~. The automatic control of the wing is a key aspect of this technology, since the optimal trajectories that the wing should fly are unstable in open loop and the system is subject to external disturbances like wind turbulence. In particular, one of the main control tasks is to make the wing fly along figure-of-eight patterns in the so-called crosswind conditions, i.e. roughly perpendicularly to the wind. This problem has been addressed by several researchers in the last years, leading to a number of contributions Canale et al. [2007], Williams et al. [2008], Baayen and Ockels [2012], Houska and Diehl [2007], Ilzhöfer et al. [2007], Fagiano et al. [2013a,b], Canale et al. [2010], Fagiano et al. [2010]. Most of these techniques employ a standard two-phase approach where a dynamical model of the system is firstly derived and then a feedback controller is designed on the basis of such a model. One of the main potential difficulties in such approaches is that the derivation of a dynamical model suited for control design might be not easy, due to the mentioned open-loop instability and the presence of disturbances which make the use of model identification techniques hard. In this paper, we show how this control problem can be tackled effectively with a direct technique, i.e. an approach that aims to design a feedback controller directly from measured data, without the need for an explicit model of the system's dynamics. In particular, we present an approach to learn the behavior of a human operator (or more in general of an existing, unknown feedback controller) using a finite number of data collected in an initial experiment. We present a theoretical analysis showing that the approach provides stability guarantees as the number of employed data-points increases, and we present the successful experimental application of the approach on a small-scale prototype built at the University of California, Santa Barbara.

⋆ 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".

2. SYSTEM DESCRIPTION

We consider a flexible wing connected by three lines to a ground unit (GU), as shown in Fig. 1.

The wing's trajectory evolves downwind with respect to the GU. For simplicity, we assume that the nominal wind direction (i.e. neglecting turbulence and small, zero-mean deviations) is aligned with the longitudinal symmetry axis of the GU, denoted by X. The latter, together with the Z axis being perpendicular to the ground and pointing upwards and with the Y axis to complete a right-handed system, forms the inertial frame $G \doteq (X, Y, Z)$, centered at the GU (see Fig. 2). By 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. 2 (dashed lines).

The wing's position $\boldsymbol{p}(t)$ can be expressed in the inertial frame G by using the spherical coordinates $\theta(t)$, $\phi(t)$ (see Fig. 2), 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. 2). 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 Fagiano et al. [2013b] for a complete derivation.

Fig. 1. Small-scale prototype built at the University of California, Santa Barbara, to study the control of tethered wings for airborne wind energy.

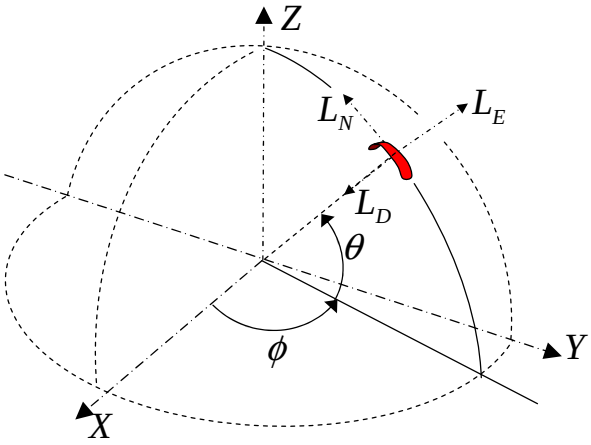

Fig. 2. Reference system $G = (X, Y, Z)$, wind window (dashed lines), variables θ, ϕ, and local north, east and down (L_N, L_E, L_D) axes.

The wing velocity vector can be expressed in the L frame as

$$\frac{d}{dt}\boldsymbol{p}(t) = r \begin{pmatrix} \dot{\theta} \\ \cos(\theta)\dot{\phi} \\ 0 \end{pmatrix}. \tag{1}$$

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

$$\gamma(t) \doteq \arctan_2\left(\cos(\theta(t))\dot{\phi}, \dot{\theta}\right), \tag{2}$$

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, $\boldsymbol{v}(t)$, 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 (see Fig. 1) and they are used to influence its trajectory: a shorter left steering line with respect to the right one impresses a left turn to the wing (i.e. a counter-clockwise turn as seen from the GU), and vice-versa. In the considered prototype, a single motor, together with a linear motion system (visible in the lower-left corner of Fig. 1), is 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 position: such a position 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 Fagiano et al. [2013a] 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 X axis (i.e. 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. Houska and Diehl [2007], Canale et al. [2010], Fagiano and Milanese [2012].

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 Houska and Diehl [2007], Williams et al. [2008], Canale et al. [2010]. 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. However, 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, a possible approach is to design a feedback controller by learning the behavior of such a human operator from experimental data. In the next section, we present a technique able to achieve this goal.

3. LEARNING A CONTROLLER FROM DATA: THEORY AND COMPUTATION

3.1 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. Both the system and 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 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:

$$\begin{aligned} u(t) &= \kappa(y(t)) \\ \kappa &: Y \to U. \end{aligned} \tag{3}$$

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, (b) neglected and time-varying dynamics, and (c) 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:

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

Remark 1. Equation (4) can be seen as a "generalized state equation". Indeed, if the output $y(t)$ includes the system's state, then equation (4) 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 (4) holds (see e.g. the controllers $\hat{\kappa}^1$ and $\hat{\kappa}^4$ in Section 4). In 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 an important 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 4.~~ □

Let us now introduce three assumptions on functions f and κ. In the following, we will make use of the function sets $\mathcal{K}$ and $\mathcal{KL}$: to this end, 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

$$\begin{aligned} &\exists \gamma_f \in (0, +\infty) : \forall e_s \in E_s, \forall y \in Y, \forall u^1, u^2 \in U, \\ &\|f(y, u^1, e_s) - f(y, u^2, e_s)\|_\infty \leq \gamma_f |u^1 - u^2|. \end{aligned} \tag{5}$$

□

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

Assumptions 1-2 imply that the closed loop system:

$$\begin{aligned} y(t+1) &= g(y(t), e_s(t)) \doteq f(y(t), \kappa(y(t)), e_s(t)) \\ g &: Y \times E_s \to Y \end{aligned} \tag{6}$$

is also described by a Lipschitz continuous function g. In particular, by construction, the function g enjoys the following properties:

$$\begin{aligned} &\exists \gamma_{g,y} \in (0, +\infty) : \forall e_s \in E_s, \forall y^1, y^2 \in Y, \\ &\|g(y^1, e_s) - g(y^2, e_s)\|_\infty \leq \gamma_{g,y} \|y^1 - y^2\|_\infty, \end{aligned} \tag{7}$$

$$\begin{aligned} &\exists \gamma_{g,e} \in (0, +\infty) : \forall y \in Y, \forall e_s^1, e_s^2 \in E_s, \\ &\|g(y, e_s^1) - F(g, e_s^2)\|_\infty \leq \gamma_{g,e} \|e_s^1 - e_s^2\|_\infty. \end{aligned} \tag{8}$$

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. We denote with $g_0 \doteq g(0,0)$ the value of g evaluated at $y = 0$, $e_s = 0$. The properties of the closed-loop system clearly depend on the controller κ, which is assumed to be stabilizing. In particular, we consider the following notion of stability:

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:

$$\forall t \geq 0,\ \|y(t)\|_\infty \leq \alpha(\|\mathbf{e}_s\|_\infty) + \beta(\|y(0)\|_\infty, t) + \delta. \tag{9}$$

□

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 $\|\mathbf{v}\|_\infty \doteq \max_{t\geq 0} \|v(t)\|_\infty$ is the $\ell_\infty-$norm of the signal $\mathbf{v}$ with the underlying norm taken to be the vector $\infty-$norm $\|v\|_\infty$.
The stabilizing properties of κ are formalized by the following assumption:

Assumption 3. The functions κ and f are such that property (7) holds with $\gamma_{g,y}(x) < 1$. □

Assumption 3 implies that the closed-loop system (6) enjoys finite-gain ℓ_∞ stability as given in Definition 1, in particular we have:

$$\begin{aligned} \forall t \geq 0,\ \|y(t)\|_\infty \leq \underbrace{\frac{\gamma_{g,e}}{1-\gamma_{g,y}} \|\mathbf{e}_s\|_\infty}_{\alpha(\|\mathbf{e}_s\|_\infty)} + \underbrace{\gamma_{g,y}^t \|y(0)\|_\infty}_{\beta(\|y(0)\|_\infty, t)} + \\ \underbrace{\frac{1}{1-\gamma_{g,y}} \|g_0\|_\infty}_{\delta}, \end{aligned} \tag{10}$$

see the Appendix for a derivation of this inequality.

Overall, Assumptions 1-3 are 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 $\tilde{u}(k)$, $\tilde{y}(k)$, $k = 0, \ldots, N-1$, 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:

$$\begin{aligned} \tilde{u}(t) &= u(t) + e_u(t) \\ \tilde{y}(t) &= y(t) + e_y(t). \end{aligned} \tag{11}$$

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 $B_\rho \doteq \{q \in \mathbb{R}^{n_q} : \|q\|_\infty \leq \rho\}$:

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

$$\tilde{u}(k) = \kappa(\tilde{y}(k)) + d(k),\ k = 0, \ldots, N-1$$

where $d(k)$ accounts for the noises $e_u(t)$ and $e_y(t)$ in (11). Since $e_u(t)$ and $e_y(t)$ are bounded and κ is Lipschitz continuous, it follows that $d(k)$ is also bounded:

$$d(k) \in B_\varepsilon, \ \forall k \geq 0. \tag{12}$$

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

Assumption 5. The set of points $\mathcal{D}_{yd}^N \doteq \{(\widetilde{y}(k), d(k))\}_{k=0}^{N-1}$ 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 value of $N_\lambda \in \mathbb{N}$, $N_\lambda < \infty$ and a pair $(\widetilde{y}(k), d(k)) \in \mathcal{D}_{yd}^{N_\lambda}$ exist such that $\|(y, d) - (\widetilde{y}(k), d(k))\|_\infty \leq \lambda$. □

Assumption 5 essentially ensures that the controller domain Y is "well explored" by the data $\widetilde{y}(k)$ and, at the same time, the noise $d(k)$ covers its domain B_ε, hitting the bounds $-\varepsilon$ and ε with arbitrary closeness after a sufficiently long time. This latter noise property is called tightness, see Novara et al. [2013] and, for a probabilistic version, Bai et al. [1998].

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) asymptotically, i.e. as $N \to \infty$, $\hat{\kappa}$ 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 2. The control design problem considered here is different from the one considered in Novara et al. [2013], where the aim of control is to track reference sequences belonging to a certain signal set. As a consequence, the stability analysis, design algorithm and asympotic properties presented in the following are different from those in Novara et al. [2013]. □

3.2 Learning algorithm

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 asymptotically the derived stability condition.

The controller $\hat{\kappa}$ is chosen to be a Lipschitz continuous function over the compact Y. Let us define the error function $\Delta : Y \to \mathbb{R}$:

$$\Delta(y) \doteq \kappa(y) - \hat{\kappa}(y). \tag{13}$$

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

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

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

$$\begin{aligned} \forall t \geq 0, \ \|y(t)\|_\infty \leq \underbrace{\frac{\gamma_{g,e}}{1 - \gamma} \|\mathbf{e}_s\|_\infty}_{\alpha(\|\mathbf{e}_s\|_\infty)} + \underbrace{\gamma^t \|y(0)\|_\infty}_{\beta(\|y(0)\|_\infty, t)} \\ + \underbrace{\frac{1}{1 - \gamma} \left(\|g_0\|_\infty + \gamma_f |\Delta_0| + \gamma_f \gamma_\kappa \varepsilon_y \right)}_{\delta}, \end{aligned} \tag{16}$$

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

Proof. See Fagiano and Novara [2013]. □

This result serves as a theoretical justification of the learning algorithm that we present in the following, which indeed is able to satisfy condition (15) in the limit, hence providing a solution to **Problem** 1.

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. Controller learning.

(1) Take a set of basis functions $\{\varphi_i\}_{i=1}^M$. The choice of this set can be carried out by means of the procedure in Fagiano and Novara [2013].
(2) Using the data set $\mathcal{D}^N \doteq \{\tilde{u}(k), \tilde{y}(k)\}_{k=0}^{N-1}$ and the basis functions chosen at step 1), define the following quantities:

$$\Phi \doteq \begin{bmatrix} \varphi_1(\widetilde{y}(0)) & \cdots & \varphi_M(\widetilde{y}(0)) \\ \vdots & \ddots & \vdots \\ \varphi_1(\widetilde{y}(N-1)) & \cdots & \varphi_M(\widetilde{y}(N-1)) \end{bmatrix} \in \mathbb{R}^{N \times M}$$
$$\widetilde{\boldsymbol{u}} \doteq (\widetilde{u}(0), \ldots, \widetilde{u}(N-1)) \in \mathbb{R}^{N \times 1}.$$

(3) Using the procedure in Fagiano and Novara [2013], obtain an estimate $\hat{\varepsilon}$ of the noise bound ε in (12), and estimates $\hat{\gamma}_f$ and $\hat{\gamma}_{g,y}$ of the Lipschitz constants γ_f and $\gamma_{g,y}$ in (5) and (7). Choose $\gamma'_\Delta \simeq (1 - \hat{\gamma}_{g,y}) / \hat{\gamma}_f$ such that $\gamma'_\Delta < (1 - \hat{\gamma}_{g,y}) / \hat{\gamma}_f$.
(4) Solve the following convex optimization problem:

$$\begin{aligned} &a^1 = \arg \min_{a \in \mathbb{R}^M} \|a\|_1 \\ &\text{subject to} \\ &(a) \ \ \|\widetilde{\boldsymbol{u}} - \Phi a\|_\infty \leq \alpha \hat{\varepsilon} \\ &(b) \ \ |\widetilde{u}(l) - \widetilde{u}(k) + (\Phi_k^r - \Phi_l^r) a| \leq \\ &\gamma'_\Delta \|\widetilde{y}(l) - \widetilde{y}(k)\|_\infty + 2\hat{\varepsilon}, \begin{cases} l = 0, \ldots, N-1 \\ k = l+1, \ldots, N-1 \end{cases} \end{aligned} \tag{18}$$

where $\Phi_k^r \doteq [\, \varphi_1(\widetilde{y}(k)) \ \cdots \ \varphi_M(\widetilde{y}(k)) \,]$ and $\alpha \geq 1$ is a number slightly larger than the minimum value for which the constraint (a) is feasible.
(5) Obtain the coefficient vector $\hat{a} = (\hat{a}_1, \ldots, \hat{a}_M)$ from the following convex optimization problem:

$$
\begin{aligned}
&(\hat{a}, \gamma_\Delta^s) = \arg \min_{a \in \mathbb{R}^M,\, \gamma''_\Delta \in \mathbb{R}^+} \gamma''_\Delta \\
&\text{subject to} \\
&(a)\ \left\| \widetilde{\boldsymbol{u}} - \Phi a \right\|_\infty \leq \alpha \hat{\varepsilon} \\
&(b)\ \left| \widetilde{u}(l) - \widetilde{u}(k) + (\Phi_k^r - \Phi_l^r)\, a \right| \leq \\
&\gamma''_\Delta \left\| \widetilde{y}(l) - \widetilde{y}(k) \right\|_\infty + 2\hat{\varepsilon},\ \begin{cases} l = 0, \ldots, N-1 \\ k = l+1, \ldots, N-1 \end{cases} \\
&(c)\ a_i = 0,\ \forall i \notin \operatorname{supp}\left(a^1\right)
\end{aligned}
\tag{19}
$$

where $\operatorname{supp}\left(a^1\right)$ 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 2 below. Step 5) aims at reducing the Lipschitz constant of the error function, maintaining the same sparsity level obtained in step 4), and satisfying the constraints for closed-loop stability. Indeed, the magnitude of this constant is linked to the maximal deviation from the trajectory achieved by the unknown controller κ, see Fagiano and Novara [2013], hence 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 nice 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.

3.3 Convergence analysis

The asymptotic properties of Algorithm 1 are analyzed here. In particular, a result is presented, showing that the controller $\hat{\kappa}$ identified by means of Algorithm 1 satisfies the stability condition (15) when the number of data N tends to infinity. Before stating the result, we need to introduce two technical assumptions. Consider the signals $v_f(k)$ and $v_g(k)$ defined as

$$
\begin{aligned}
v_f(k) &\doteq f(y(k), u(k), e_s(k)) - f(y^*, \tilde{u}(k), e_s^*) + e_y(k+1) \\
v_g(k) &\doteq g(y(k), e_s(k)) - g(\tilde{y}(k), e_s^*) + e_y(k+1).
\end{aligned}
$$

Assumption 6. The set of points $\mathcal{D}_{uv}^N \doteq \{\widetilde{u}(k), v_f(k)\}_{k=0}^{N-1}$ is dense on $U \times B_{\varepsilon_f}$ as $N \to \infty$. □

Assumption 7. The set of points $\mathcal{D}_{yv}^N \doteq \{(\widetilde{y}(k), v_g(k))\}_{k=0}^{N-1}$ is dense on $Y \times B_{\varepsilon_g}$ as $N \to \infty$. □

In Assumptions 6-7, density of the sets $\mathcal{D}_{uv}^N$, $\mathcal{D}_{yv}^N$ is intended in the same sense as in Assumption 5.

Theorem 2. Let the optimization problem (18) be feasible for any $N \geq 0$. Let Assumptions 1-2, 4, and 5-7 hold. Then, the error function $\Delta \doteq \kappa - \hat{\kappa}$ is Lipschitz continuous on Y, with constant γ_Δ such that

$$
\limsup_{N \to \infty} \gamma_\Delta \leq \gamma_\Delta^s < \frac{1 - \gamma_{g,y}}{\gamma_f}.
$$

Proof. See Fagiano and Novara [2013]. □

4. EXPERIMENTAL RESULTS

4.1 Data collection and controller learning

We have considered θ, ϕ, $\dot{\theta}$, $\dot{\phi}$, and γ as candidates feedback variables, since they can be measured or estimated with good accuracy and they provide a representation of the quantities that the human operator observes when controlling the wing. In particular, we present here the results obtained by considering 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 1 shows the choice of output vector y for these four cases. The employed sampling time is $T_s = 0.02\,\mathrm{s}$. Albeit also 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. Fig. 3 shows a typical full cycle, i.e. a single 8-shaped trajectory, in the ϕ, θ plane, together with the related courses of $\dot{\phi}, \dot{\theta}$, of the velocity angle γ, and of the control input u. As it can be noted, the period of a single cycle is about 5 s, hence a total of about 130 cycles has been used for the design of the four controllers. Fig. 4 shows the course of the wind speed during the time when the identification data were collected. The average wind speed was 5.44 m/s.

We have chosen polynomial basis functions of the following form:

$$
\varphi_i(y) = y_{j_1}^\alpha y_{j_2}^\beta;\, j_1, j_2 = 1, \ldots, n_y;\, \alpha, \beta = 0, \ldots, 3. \tag{20}
$$

For each combination of j_1 , j_2, α , β in (20), we assigned a progressive index $i = 1, \ldots, M$ to the corresponding polynomial, where $M = (q\, n_y + 1)(q\, n_y + 2)/2$. We estimated the values of the design parameters $\hat{\varepsilon}$, $\hat{\gamma}_{g,y}$ and$\hat{\gamma}_f$, involved in Algorithm (1), by means of the algorithms presented in Subsection 3.2. The corresponding number of optimization variables in problems (18)-(19) is indicated in Table 1, too, together with the times required for the learning algorithm to derive the different controllers and the related numbers of resulting non-zero components. It can be noted that, starting from a quite large number of decision variables, the approach yields control laws with few non-zero terms, which we implemented on a real-time machine using the xPC Target® toolbox of Matlab®. 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 Grant and Boyd [2010]) are quite low. Indeed, a considerable number of data have been used for design: $N = 10501$. The data with indexes $l = 1, 30, 60, \ldots, N-1$ 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. As regards the times required for the online control computation, these 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.

4.2 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. We denote with $\overline{\theta}_k$, $\overline{\phi}_k$ the average position, in

Table 1. Characteristics of the learned controllers.

Controller	Feedback variables y	N. of optim. variables	N. of non-zero terms	Comp. time (s)
$\hat{\kappa}^1$	$\theta(t)\ \ \theta(t-50)$ $\phi(t)\ \ \phi(t-50)$ $\dot{\theta}(t)\ \ \dot{\theta}(t-50)$ $\dot{\phi}(t)\ \ \dot{\phi}(t-50)$ $\gamma(t)\ \ \gamma(t-50)$	496	38	1025
$\hat{\kappa}^2$	$\theta(t)$ $\phi(t)$ $\dot{\theta}(t)$ $\dot{\phi}(t)$ $\gamma(t)$	136	41	458
$\hat{\kappa}^3$	$\theta(t)$ $\phi(t)$ $\dot{\theta}(t)$ $\dot{\phi}(t)$	91	36	90
$\hat{\kappa}^4$	$\theta(t)\ \ \theta(t-50)$ $\phi(t)\ \ \phi(t-50)$ $\dot{\theta}(t)\ \ \dot{\theta}(t-50)$ $\dot{\phi}(t)\ \ \dot{\phi}(t-50)$	325	45	505

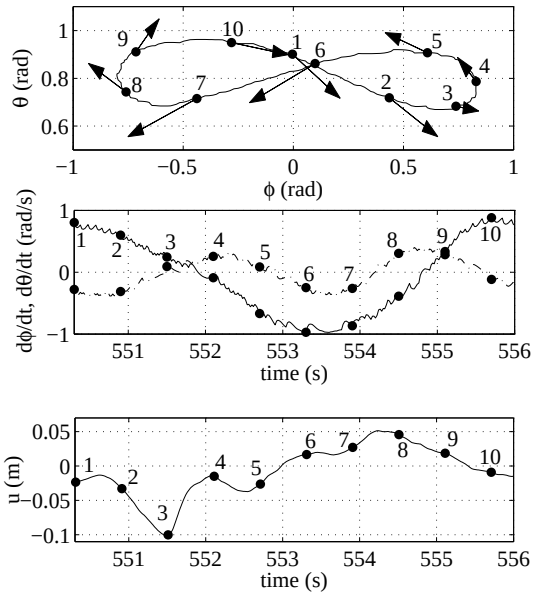

Fig. 3. Sample figure-eight trajectory during manual flight. From top: wing's path in ϕ, θ coordinates and related gradient estimated by the velocity angle γ (arrows); angular velocities $\dot{\phi}$ (solid) and $\dot{\theta}$ (dash-dot); control input u.

spherical coordinates, of the $k-$th 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, we computed the following quantities:

$$\begin{aligned} &\overline{\phi} \doteq \frac{1}{K}\sum_{k=1}^{K}\overline{\phi}_k, && \overline{\theta} \doteq \frac{1}{K}\sum_{k=1}^{K}\overline{\theta}_k \\ &\Delta\phi \doteq \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(\overline{\phi}_k-\overline{\phi})^2}, && \Delta\theta \doteq \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(\overline{\theta}_k-\overline{\theta})^2} \\ &\overline{\Delta\phi} \doteq \max_{k=1,\dots,K}|\overline{\phi}_k-\overline{\phi}|, && \overline{\Delta\theta} \doteq \max_{k=1,\dots,K}|\overline{\theta}_k-\overline{\theta}|. \end{aligned} \tag{21}$$

The variables (21) 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

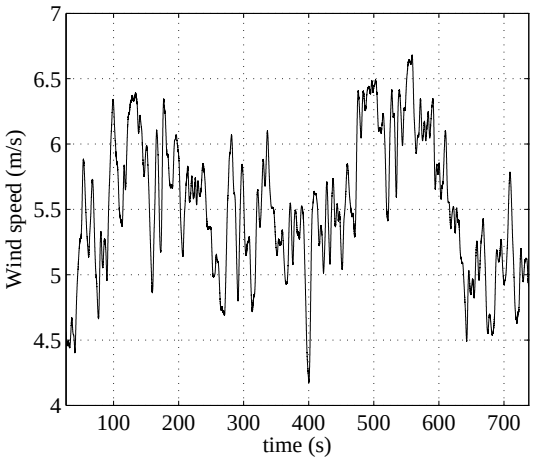

Fig. 4. Wind speed measured at 4 m above the ground during the manual flight.

Table 2. 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

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 Zgraggen et al. [2013]. 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 Zgraggen et al. [2013].

The results obtained with the four designed controllers are reported in Table 2. 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. 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 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 2. The wind speed can be considered as an exogenous, unmeasured disturbance, which can be embedded in the variable $e_s(t)$ of (4). 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 Subsection 3.2 and with physical considerations on the system. The controller $\hat{\kappa}^4$ was able to achieve figure-eight trajectories, however

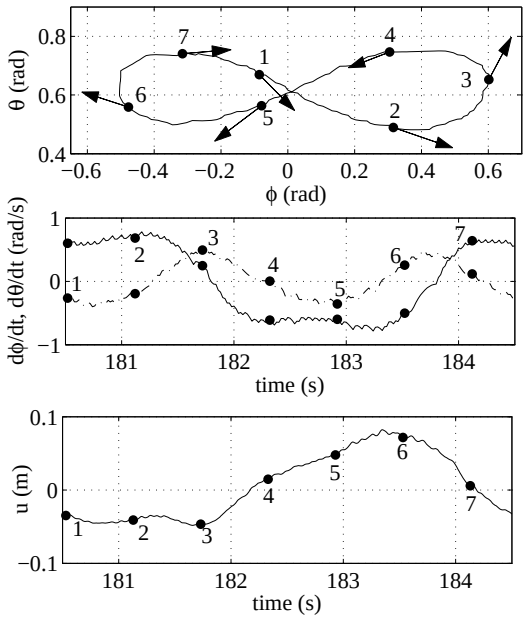

Fig. 5. Experimental results for the learned controller $\hat{\kappa}^2$ and. From top: wing's path in ϕ, θ coordinates and related gradient estimated by the velocity angle γ (arrows); angular velocities $\dot{\phi}$ (solid) and $\dot{\theta}$ (dash-dot); control input u.

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 Fagiano [2012]). 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 (2), is not described exactly by the polynomials (20), 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 in this specific case), 3) the use of ℓ_1-norm minimization allows the control designer to start with a relatively large number of candidate feedback variables, leaving to the algorithm the task to identify the most relevant ones. 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. From Table 2, 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.

5. CONCLUSIONS

We presented an approach to design a feedback controller that stabilizes crosswind flight patterns of a tethered wing. The proposed technique is able to derive a controller directly from the data measured on an existing feedback control system, where both the controller and the plant are not known. 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 leads 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 converge to a stabilizing controller as the number of employed data points tends to infinity. We presented the experimental application of the approach on a small-scale prototype, showing promising results.

REFERENCES

U. Ahrens, M. Diehl, and R. Schmehl (editors). *Airborne Wind Energy*. Springer, 2013.

J. H. Baayen and W. J. Ockels. Tracking control with adaption of kites. *IET Control Theory and Applications*, 6(2):182–191, 2012.

E.W. Bai, H. Cho, and R. Tempo. Convergence properties of the membership set. *Automatica*, 34(10):1245–1249, 1998.

M. Canale, L. Fagiano, and M. Milanese. Power kites for wind energy generation. *IEEE Control Systems Magazine*, 27(6):25–38, 2007.

M. Canale, L. Fagiano, and M. Milanese. High altitude wind energy generation using controlled power kites. *IEEE Transactions on Control Systems Technology*, 18(2):279 –293, mar. 2010. ISSN 1063-6536. doi: 10.1109/TCST.2009.2017933.

L. Fagiano. *Learning a nonlinear controller from data. Experimental test movie. Available online: http://lorenzofagiano.altervista.org/movies/LC_kite_LD.mp4.* 2012.

L. Fagiano and M. Milanese. Airborne wind energy: an overview. In *American Control Conference 2012*, pages 3132–3143, Montreal, Canada, 2012.

L. Fagiano and C. Novara. Identification of nonlinear controllers from data: theory and computation. In *arXiv:1309.1574 [cs.SY]*, 2013.

L. Fagiano, M. Milanese, and D. Piga. High-altitude wind power generation. *IEEE Transactions on Energy Conversion*, 25(1):168 –180, mar. 2010. ISSN 0885-8969. doi: 10.1109/TEC.2009.2032582.

L. Fagiano, K. Huynh, B. Bamieh, and M. Khammash. On sensor fusion for airborne wind energy systems. *IEEE Transactions on Control Systems Technology*, to appear, available online, 2013a.

L. Fagiano, A.U. 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*, in press, online version available, 2013b.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, August 2010.

B. Houska and M. Diehl. Optimal control for power generating kites. In *9^{th} European Control Conference*, pages 3560–3567, Kos, GR, 2007.

A. Ilzhöfer, B. Houska, and M. Diehl. Nonlinear MPC of kites under varying wind conditions for a new class of large-scale wind power generators. *International Journal of Robust and Nonlinear Control*, 17:1590–1599, 2007.

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

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

A.U. Zgraggen, L. Fagiano, and M. Morari. On real-time optimization and adaptation of airborne wind energy generators. In *accepted for the IEEE Conference on Decision and Control 2013*, Firenze, Italy, 2013.