Improved Path Following for Kites with Input Delay Compensation

Nikitas Rontsis[1], Sean Costello[2], Ioannis Lymperopoulos[2] and Colin N. Jones[2],

Abstract— In kite power systems, substantial input delay between the actuator and the tethered kite can severely hinder the performance of the control algorithm, limiting the capability of the system to track power-optimal loops. We propose a method that deals with this impediment by using a data-based adaptive filter that predicts future states despite variations in wind conditions, other exogenous disturbances and model mismatch. Moreover, we exploit the geometry of the path on a hemisphere to enhance the guidance algorithm for such kites at a fixed length tether. The objective is to improve the automatic crosswind operation of an airborne wind energy system. To test this under realistic conditions, a small-scale prototype was employed for a series of experiments. The robustness to disturbances and the performance of the algorithm in path following was evaluated for a number of different paths.

I. INTRODUCTION

This paper considers the control of power-generating kites. Harnessing wind power using kites is a radical new renewable-energy concept that is currently being developed (for a review of this rapidly expanding field see [1] and [2]). Kites can access strong and consistent winds at altitudes out of reach of wind turbines. Moreover, they do not require a large supporting tower. Kites are essentially flying wings (attached to the ground by a flexible tether), ranging from flexible para-glider type designs to rigid composite aircraft wings.

Most of the AWE prototype systems, and the ones currently in development, are operating between 50-200 meters [1], [2]. Future deployment that could reach up to 500-1000 meters would enjoy a wind power density that is up to four times higher [3]. A kite flying roughly perpendicular to the wind travels at very high speed and experiences large aerodynamic forces. This force is transmitted to the ground via the tether, where it can be used to drive a generator for electricity production, by slowly unreeling the tether. In a second phase, the kite is reeled back in (using a small percentage of the generated electricity) while it is not flying cross-wind, thus generating very little tether force. This is known as pumping-cycle operation [4]. Other than electricity production, kites can be used in various interesting applications. For example large kites are being used to propel cargo ships [5].

A number of technical barriers must be crossed in order to achieve reliable kite power systems. One of the most fundamental is automatic control of the kite operation. The type of kite used for power generation is inherently unstable, which means that without continuous input steering, it will crash in seconds. Moreover, a kite can be given different paths to follow, and it is this path that determines how much power is produced under specific wind conditions. The challenge is to design *path-following* control algorithms capable not only of keeping the kite flying, but also of following a specific power-optimal path. The control task is complicated to a large extent by a number of *time delays* affecting the system. If a kite is actuated from the ground, the elasticity of the long lines introduces a non-negligible delay between line adjustments on the ground and the resulting steering deflection experienced by the kite. Even if the kite has on-board actuators, data-transmission delays, actuator response times, and filter-induced delay are unavoidably present. As the kite reaches speeds in excess of 150 km/h, even time delays of a fraction of a second can seriously compromise controller performance.

Initial development in the field of kite control focused on advanced, model-based control, including nonlinear model predictive control (NMPC) using an economic [6] or tracking [7], [8] objective, and Lyapunov redesign techniques [9]. However, there is still no reported experimental implementation of an NMPC controller for kites, probably because of the inaccuracy of existing kite models, the considerable computational effort and the added complexity of the method. In contrast, a number of experimentally-validated simple geometric control algorithms have recently been published. In particular, [5] devised and extensively tested a bang-bang controller for achieving figure-of-eights (the path shape that is commonly used for power production), based on an ingeniously simple kinetic model. [10] validated a similar cascade-control strategy on a small-scale prototype, which alternately directs the kite towards one of two target points, again producing a figure-of-eight flight path. Both of these approaches allow the kite's path to be adjusted to a limited extent. Both [11] and [4] recently published preliminary experimental results on the implementation of path-following controllers. While reasonable path-following performance was obtained, [11] identified time delay as one of the main barriers to improving path-following control performance.

This paper presents a path following controller for kites, specifically designed to cope with significant time delays. The main novelty is a prediction algorithm that computes the kite's future states in order to counteract delay, using an adaptive turn-rate law. The adaptive turn-rate law is updated at each sampling period using experimental data, allowing it to adapt to variations in the wind conditions or the kite's

[1]Nikitas Rontsis is with the Department of Electrical & Computer Engineering, Aristotle University of Thessaloniki, 54124, Thessaloniki, Greece `nikitas@rontsis.com`

[2]Sean Costello, Ioannis Lymperopoulos and Colin N. Jones are with the Department of Mechanical Engineering, Automatic Control Laboratory, EPFL, 1015 Lausanne, Switzerland `sean.costello@epfl.ch`, `ioannis.lymperopoulos@epfl.ch`, `colin.jones@epfl.ch`

parameters. The Non-Linear Guidance Law (NLGL) [12] is used to calculate the steering inputs required to follow the reference path. This guidance law, which was originally developed for constant-curvature paths, is generalized in this paper to follow figure-of eight paths. The algorithm is successfully applied to a small-scale experimental kite system. The results show that, although path-following is not perfect, the kite robustly follows arbitrary figure-of-eight paths, despite relatively long time delays.

The paper is structured as follows: Section II reviews a simple dynamic model for kites. Section III presents the controller algorithm, while Section IV presents the experimental validation. Finally, Section V concludes and discusses future work.

II. Model Equations

To capture the kite behavior we use a simple three state model originally proposed by [5]. The equations correspond to a flexible kite, similar to those used in kite surfing, with a constant-length tether. In addition to neglecting inertial effects, the model assumes a constant homogeneous wind field and a constant lift-to-drag (glide) ratio E for the kite.

The model of the kite is comprised of the following ordinary differential equations:

$$\dot{\theta} = \frac{\bar{w}}{L}\left(\cos\psi - \frac{\tan\theta}{E}\right), \qquad (1a)$$

$$\dot{\phi} = -\frac{\bar{w}}{L\sin\theta}\sin\psi, \qquad (1b)$$

$$\dot{\psi} = \bar{w}gu(t-\tau) + \dot{\phi}\cos\theta, \qquad (1c)$$

where the angles θ and ϕ determine the position of the kite using the spherical coordinate system explained in Appendix, and ψ represents its orientation, which can be represented by a scalar, given that the yaw axis of the kite always remains parallel to the kite's position vector [5]. E is the glide ratio, L is the line length, and g is the steering coefficient. The system input, u, which is the steering deflection applied to the kite, is considered in this work to be subject to a time delay, τ. The airflow parallel to the kite's roll axis, $\bar{w}$, is given by

$$\bar{w} = wE\cos\theta, \qquad (2)$$

based on the assumption that the wind vector is parallel to $\mathbf{e}_{\mathrm{x}}$ (Fig. 16) with magnitude w.

An important variable for the control of the kite, which has been used for the derivation of geometrical control algorithms [11], [10], is the kite's velocity angle, denoted γ. Given that the kite position lies on a sphere, its velocity $\boldsymbol{v}$ is on the plane that is tangent to the sphere at the kite's position. Thus, we can decompose $\boldsymbol{v}$ into its components in the $\mathbf{e}_{\phi}$ and $\mathbf{e}_{\theta}$ directions (see Fig. 16). The angle between $\boldsymbol{v}_{\phi}$ and $\boldsymbol{v}$ is the velocity angle:

$$\gamma = \tan^{-1}\left(\frac{\dot{\phi}\sin\theta}{\dot{\theta}}\right) = \tan^{-1}\left(\frac{\sin\psi}{\cos\psi - \tan\theta/E}\right). \quad (3)$$

III. Controller

We propose the hierarchical control schema shown in Fig. 1. The controller objective is to follow the desired path. We denote as a path $\bar{\mathbf{c}}_{\mathrm{d}}(\bar{s}) : T \mapsto S \subset \mathbb{R}^3$, a closed, twice-differentiable curve parametrized by arc-length $\bar{s} \in T = [0, l_{\mathrm{d}}]$, with $\bar{\mathbf{c}}_{\mathrm{d}}(0) = \bar{\mathbf{c}}_{\mathrm{d}}(l_{\mathrm{d}})$ with l_{d} denoting the total length of the curve, and S the surface of a unit sphere.

Practically, we are only interested in paths that can be tracked by our physical system. According to the model (1a)-(1c), this implies paths with bounded geodesic curvature $\kappa_{\mathrm{g}}(\bar{s})$, that is $|\kappa_{\mathrm{g}}(\bar{s})| < \kappa_{\max}$, where $\kappa_{\max}$ depends on the maximum allowable steering input.

In order to avoid discontinuity problems introduced by the start $\bar{s} = 0$ and the end $\bar{s} = l_d$, we define a periodic format of the desired path $\mathbf{c}_{\mathrm{d}}(s) : \mathbb{R} \mapsto \mathbb{R}^3$ which essentially extends the path by repeating it infinitely many times:

$$\mathbf{c}_{\mathrm{d}}(s) = \bar{\mathbf{c}}_{\mathrm{d}}\left(s - \left\lfloor\frac{s}{l_{\mathrm{d}}}\right\rfloor l_{\mathrm{d}}\right), \quad s \in \mathbb{R}, \qquad (4)$$

with $\lfloor\cdot\rfloor$ denoting the floor function.

A. Delay Compensation

A compensation mechanism for the input delay τ is essential, because of its considerable effect on the stability and performance of the control algorithm. However, delay compensation requires a well-calibrated system model. For this reason, an adaptive turn-rate law, based on measurements is developed.

1) Adaptive Filter: By reordering Eq. (1c), we derive the corrected turning rate $\bar{\dot{\psi}}_k$:

$$\bar{\dot{\psi}}_k := \dot{\psi}_k - \dot{\phi}_k\cos\theta_k = \bar{w}_k g u_{k-N_{\mathrm{d}}}, \qquad (5)$$

where $u_{k-N_{\mathrm{d}}}$ is the input $N_{\mathrm{d}} := \lfloor\tau/T_{\mathrm{s}}\rfloor$ sampling periods in the past. In this section, the subscript will denote the discrete-time sampling index with sampling period T_{s}.

There are a number of reasons why Eq. (5) is not accurate enough to be used for delay compensation, which necessitates the future states to be *predicted*. Firstly, the relationship between turning rate $\bar{\dot{\psi}}$ and the steering input u is not exactly linear, as steering occurs due to the complex non-linear asymmetric deformation of the kite [13]. Secondly, the correct turning coefficient g is usually not known, as it varies depending on the exact kite configuration and the wind speed. Thirdly, the inertia of the kite membrane indicate that the current turning rate is affected by the actuation signal over a period of time while Eq. (5) considers only the last (delayed) input command. In addition, reliable measurements of the oncoming wind speed experienced by the kite, $\bar{w}$, requiring an on-board pitot tube, are often not available. Neither is there (currently) any good method for estimating $\bar{w}$ from the wind speed measured at the ground station, w [5].

Thus, it was necessary to design a delay-compensation algorithm that does not rely on the inaccurate turning law, nor require real-time wind measurements, but can nonetheless adapt to changing flight conditions. An adaptive finite
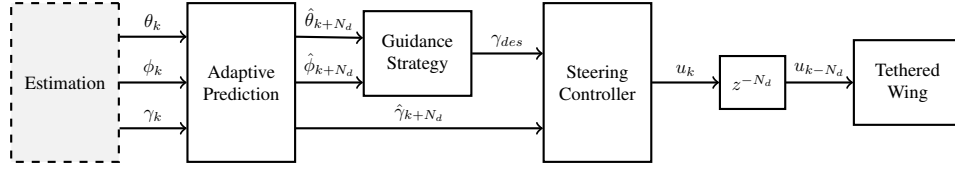
Fig. 1: Controller schema.

impulse-response filter, which also considers the effect of past inputs, was found capable of accurately modeling the kite's turning behavior:

$$\bar{\psi}_k = \mathbf{h}_k^T \underline{\mathbf{u}}_k, \tag{6}$$

where $\mathbf{h}_k \in \mathbf{R}^l$ is the vector of the weights, and $\underline{\mathbf{u}}_k = [u_{k-N_d} \cdots u_{k-N_d-l+1}]^T$ is the lifted vector of delayed inputs. The filter coefficients are adapted in real-time in an attempt to minimize the mean square error as a function of the filter coefficients. In particular, given an estimation $\hat{\mathbf{h}}_k$ of the filter coefficients $\mathbf{h}_k$, we are trying to minimize the following metric:

$$\mathcal{E} = E\left[\bar{e}_k^2\right] = E\left[\left(\bar{\psi}_k - \hat{\mathbf{h}}_k^T \underline{\mathbf{u}}_k\right)^2\right] \tag{7}$$

where $E[\cdot]$ denotes the mathematical expectation and $\bar{e}_k = \bar{\psi}_k - \hat{\mathbf{h}}_k^T \underline{\mathbf{u}}_k$ is the error signal.

For the adaptation of weights, we use the LMS (Least-Mean-Squares) algorithm. LMS is a stochastic gradient algorithm, which recursively computes the filter coefficients, based on stochastic-gradient-descent, in order to minimize $\mathcal{E}$. The update equation of LMS [14] is the following:

$$\hat{\mathbf{h}}_{k+1} = \hat{\mathbf{h}}_k + \mu \bar{e}_k \underline{\mathbf{u}}_\mathbf{k}, \tag{8}$$

where μ is the step size.

2) Prediction: Integrating Equations (1a)-(1b) and the adaptive turning law (6), with the weights obtained from Eq. (8), we obtain the predictions $\hat{\theta}_{k+N_d}, \hat{\phi}_{k+N_d}, \hat{\gamma}_{k+N_d}$ Note that Equations (1a)-(1b) require $\bar{w}$. A fixed, user-specified estimate of wind (w) is used to approximate $\bar{w}$ for this purpose, according to Eq. (2). This introduces a small, unavoidable error in the prediction of the kite's *position*. However, this does not appear to be a problem for the control algorithm, which is mostly sensitive to errors in the prediction of the kite's *orientation*. The kite's orientation is predicted very accurately, irrespective of the wind speed, thanks to the adaptive filter.

B. Guidance Strategy

A simple, commonly employed geometric algorithm was used for guidance, called the Nonlinear Guidance Law (NLGL) [12]. [15] provide a detailed definition and comparison with other guidance algorithms, and show that NLGL performs well compared to similar navigation algorithms. In short, given the desired path $\mathbf{c}_d$, NLGL determines the desired point (commonly referred to as the Virtual Target Point, VTP) with the help of an auxiliary circle of radius m located at the current position, $\mathbf{a}$, of the kite (see Fig. 2). For
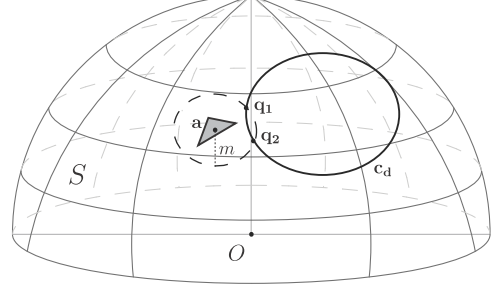

Fig. 2: The NLGL algorithm. The kite is depicted with the gray triangle.

simple scenarios (if $\mathbf{c}_d$ is a circle or a line and $\mathbf{a}$ is close enough to $\mathbf{c}_d$) the circle intersects $\mathbf{c}_d$ at two points $\mathbf{q}_1$, $\mathbf{q}_2$. Depending on the desired direction of movement along the path, we choose the appropriate point from among them.

Since the shape of the desired path is, for this problem, a figure-of-eight path, the number of intersections between $\mathbf{c}_d$ and the auxiliary small circle can be zero, one or greater than one. Thus, a few generalizations are introduced into the algorithm. The basic idea underlying them is that the current VTP should be close to the previous one, thus enforcing consistency and smoothness of the arc-length parameter corresponding to VTP.

In order to avoid multiple intersections we only consider a part of the trajectory. In particular, if s_k is the arc-length parameter of $\mathbf{c}_d(s)$ that corresponds to the virtual target point at time k, then in order to calculate s_{k+1}, we will only consider $\mathbf{c}_d(s)$, $s \in [s_k, s_k + \delta]$, where δ is a design parameter that depends on the kite's velocity as well as the desired path.

We determine s_k as follows:

1) For the initial choice s_0, we choose the point with the following arc-length parameter

$$s_0 = \underset{s\in[0,l_d)}{\arg\min}\, d_g\left(\mathbf{c}_d(s), \mathbf{p}\right), \tag{9}$$

where $\mathbf{d}_g\left(\mathbf{a}, \mathbf{b}\right)$ denotes the length of the shortest path between points $\mathbf{a}$, $\mathbf{b}$ on the sphere (geodesic distance, refer to Eq. (21)).

2) In the case of no intersections, we choose the point with the following arc-length parameter:

$$s_{k+1} = \underset{s\in[s_k,s_k+\delta]}{\arg\min}\, d_g\left(\mathbf{c}_d(s), \mathbf{p}\right). \tag{10}$$

3) If a set $Z = \{s_1', \ldots, s_N'\}$ of parameters, that correspond to different intersections, are found, then the closest one to s_k is returned:

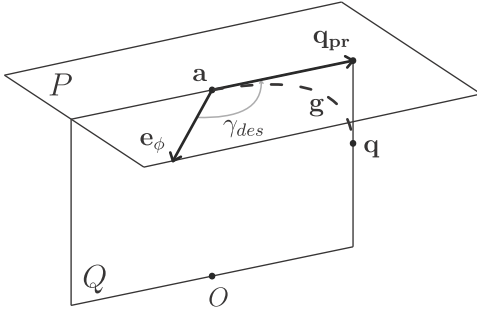$$s_{k+1} = \underset{s'\in Z}{\min}\, |s_k - s'|. \tag{11}$$

Fig. 3: Definition of γ_{des} via the geodesic from the current position of the kite $\mathbf{a}$ to the VTP $\mathbf{q}$. The point O denotes the center of the sphere upon which the motion of the kite is constrained.
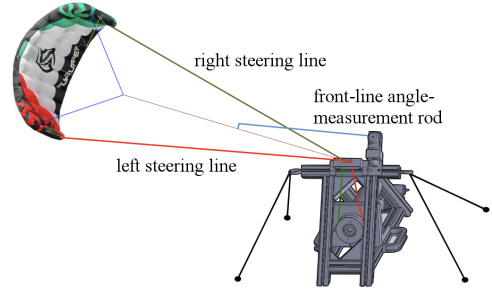

Fig. 4: Experimental Setup.

IV. EXPERIMENTAL APPLICATION

A. Physical System

The autonomous operation of flexible kites was tested on the small scale prototype system of EPFL, which is depicted in figure 4. The prototype is designed for crosswind flight with constant tether length. It can be broken into two parts: the base station and the kite. Three lines are used to connect the kite with the base station. The highly-tensioned front line transmits the aerodynamic force acting on the kite to the ground station, while the left and right lines are used to steer the kite. The input u is a difference in length between the right and left line, whose relative lengths are adjusted by a motor. The *only* measurements are the angles of the front line, from which the position of the kite, (θ, ϕ), is inferred. The kite's orientation ψ and its derivative $\dot{\psi}$ are calculated via a smooth first order and second order derivation respectively. A similar experimental platform with feedback estimation of parameters can be found in [17].

The system is small scale, in the sense that the kite is smaller and the lines are shorter than those used in current large-scale systems. According to Eq. (1a)-(1c), the rate at which the system's states evolve is inversely proportional to the line length. This makes delay a significant factor for the performance of the controller, as our experimental system's dynamics are essentially much faster than those of a large-scale system. Fortunately, the elasticity of the lines, which is the main cause of delay for large scale systems, becomes negligible for a small-scale system. However, due to the rapidity of the small-scale system, communication delay, actuator delay and delay induced by filters, contribute significantly to the overall delay. Overall, counteracting the approximately 200 ms of delay experienced by the small-scale experimental system with 25 m lines is equivalent, from a control point of view, to counteracting 2 s of delay for a system with 250 m lines.

B. System and controller parameters

Table I presents the parameters of the experimental system, while Table II presents the various parameters of the controller. A $2.5m^2$ Flysurfer Viron was used during the experiments presented here.

We estimate the delay $\tau = N_{\text{d}} \cdot T_{\text{s}}$, and proportionality constant $\bar{g} = \bar{w}g$ from a dataset according to the following

Finally, in order to further avoid inconsistencies due to measurement noise, a saturation in the evolution of the VTP is introduced:

$$s_{k+1} \in [s_k + \upsilon_{min} T_{\text{s}}, s_k + \upsilon_{max} T_{\text{s}}], \qquad (12)$$

where $\upsilon_{\min}$, $\upsilon_{\max}$ are design parameters that represent the minimum and maximum velocity of the VTP.

We define the desired velocity angle γ_{des} as the velocity angle that the kite would have if it was moving on the shortest route towards the VTP. As depicted on Fig. 3, the shortest route $\mathbf{g}$ (geodesic) from the kite position to VTP belongs to a plane Q which is perpendicular to the sphere's tangent plane P at the kite position $\mathbf{a}$ [16]. Therefore the projection of curve $\mathbf{g}$ to P is a straight line. As a result, the desired velocity angle γ_{des} can be calculated by the vector from $\mathbf{a}$ to the projection $\mathbf{q}_{\text{pr}}$ of $\mathbf{q}$ to P.

Thus, according to Eq. (3), we calculate the desired velocity angle γ_{des} as:

$$\gamma_{\text{des}} = \text{atan} \left((\mathbf{q}_{\text{pr}} - \mathbf{a}) \cdot \mathbf{e}_\phi, (\mathbf{q}_{\text{pr}} - \mathbf{a}) \cdot \mathbf{e}_\theta \right). \qquad (13)$$

C. Steering Controller

A proportional controller tries to force the kite's velocity angle to match the reference value provided by the Guidance Strategy. The controller acts on the error:

$$e_k = \gamma_{\text{des}} - \hat{\gamma}_{k+N_{\text{d}}}, \qquad (14)$$

applying the following input to the system:

$$u_k = \frac{1}{Lg} K_{\text{P}} e_k. \qquad (15)$$

The choice of the proportional controller is justified by the first-order (approximate) relationship between the input and the kite's turning rate (see Eq. (5)). In particular, it is shown in Appendix that the following equation relates the geodesic curvature κ_{g} and the input:

$$\kappa_{\text{g}} = Lgu = K_{\text{P}} e_k \qquad (16)$$

Thus, under the effect of the controller, the kite turns proportionally to the error e_k. The scaling fraction $\frac{1}{Lg}$ in Eq. (15) is introduced to cancel the dependence of κ_{g} on L and g.

equation:

$$(\bar{g}, N_{\mathrm{d}}) = \arg\min_{g', N'_{\mathrm{d}}} \left(\sum_{k=0}^{N} \left| \bar{\dot{\psi}}_k - g' u_{k-N'_{\mathrm{d}}} \right| \right) \qquad (17)$$

The reader might guess that, based on Eq. (17), the first N_{d} weights of the LMS filter should be zero. However, this is not necessarily the case. With the SkySails model (Eq. (1a)-(1c)), we have an approximate, but mathematically precise, explanation for the effect of the input via Eq. (16), and thus it is useful for controller design. N_{d} is an estimate of the delay, obtained subject to the assumptions governing the Skysails model. In reality, the delay-like behavior is caused by dynamics (either the line's or the kite's). As a result, the turning law from the SkySails model is not sufficiently accurate for the prediction step necessary to compensate delay. That is why the prediction uses the adaptive filter as a turn-rate model, which, by taking real-time data into account, can better model the kite's current *dynamic* behavior. Hence the first N_{d} values of the LMS filter will not necessarily be zero.

The glide ratio, E, is chosen based on the specification of the kite that is used. A user specified approximation of the wind magnitude w and orientation ξ are provided by the operator.

The gain K_{P} determines the aggressiveness of the controller. Larger values result in closer following, up to a point where overshoots start to occur. Although the controller was designed to be independent of g and L, larger L values lead to a slower system, thus better performance is achieved, and more aggressive gains can be used. Similarly, according to Eq. (16), the resulting trajectory is independent of the wind magnitude, but an increase in wind speed results in a faster system. The auxiliary circle radius m determines how aggressive the choice of the VTP is. Choosing $m = 0$ will result in the algorithm steering the kite perpendicularly to the desired path, which will unavoidably result in overshoot. On the other hand, large values will result in poor tracking, multiple intersections, and even unstable behavior. Finally, since both m and K_{P} determine the aggressiveness of the algorithm, these two parameters are coupled.

Parameters δ, $v_{\max}$ and $v_{\min}$ depend on the desired path, and they are tuned in order to solve the problems explained in subsection III-B. The input is saturated at the value $u_{\max}$, to avoid large inputs that could cause the kite to stall in light winds.

Finally, the input u is filtered with a 1st order Butterworth filter, with cutoff frequency f_{c}, to provide a smooth setpoint for the steering motor.

C. Evaluation of the prediction

In order to evaluate the prediction we will present the results of two different adaptive scenarios while the kite performs challenging maneuvers instructed by an operator (non-autonomous operation). We compare the results obtained with a simple proportional relationship ($l_{\mathrm{LMS}} = 1$), and with $l_{\mathrm{LMS}} = 20$, which translates to $20 \cdot T_{\mathrm{s}} = 0.6s$ look-back time.

Parameter	Explanation	Value	Units
L	Tether length	25	m
T_{s}	Sampling period	30	ms
τ	Identified delay	180	ms
E	Glide ratio	Kite specific	-
w	Wind magnitude	Variable	m/s
ξ	Wind angle	Variable	rad

TABLE I: System Parameters

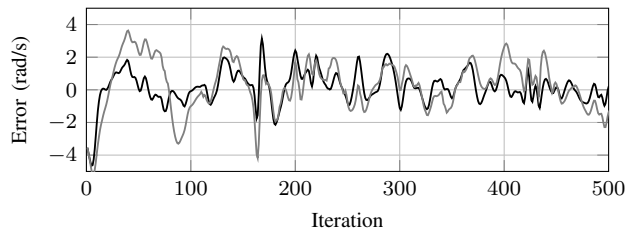| Parameter	Explanation	Value	Units
K_{P}	Proportional gain	Variable	-
m	Auxiliary circle radius	Variable	m
f_{c}	Cut off frequency	3	Hz
δ	Trajectory percentage taken into account in the calculation of VTP	$\frac{1}{7}l_{\mathrm{d}}$	-
$v_{\max}$	Max velocity of VTP	$\frac{1}{35}l_{\mathrm{d}}$	m/s
$v_{\min}$	Min velocity of VTP	0	m/s
$u_{\max}$	Max allowable input	0.3	m
μ	LMS learning rate	0.05	-
l_{LMS}	LMS length	20	-

TABLE II: Controller Parameters

Fig. 5: Error of the LMS filtering for different lookback time, as compared to the actual turning rate $\bar{\dot{\psi}}$. Results with no lookback time are shown with gray line, while results for $l_{\mathrm{LMS}} = 20$ are shown with black line. The results, after settling (iteration > 100), with lookback time are considerably better with mean$(|\mathrm{error}|) = 0.77$, max $|\mathrm{error}| = 3.51$, as compared to without lookback mean$(|\mathrm{error}|) = 1.01$, max $|\mathrm{error}| = 4.17$.

The adaptation algorithm for both filters was LMS, with step size $\mu = 0.2$. The filters are initialized with zero weights at iteration $k = 0$. As we can observe from figure 5, the filter with $l_{\mathrm{LMS}} = 20$ performs considerably better than the simple proportional filter with $l_{\mathrm{LMS}} = 1$, while at the same time its output converges faster to the desired $\bar{\dot{\psi}}$.

Despite these promising results, one should be aware of the limitations of the LMS prediction. The weights of the filter do not reach an equilibrium but change over time, which means that one might need to be conservative with the step size. Moreover, in order to handle more noisy environments than the one presented here, some adaptations can be introduced, such as Block LMS [14], which, however, might make the convergence slower.

D. Effects of the prediction

In order to illustrate the importance of the delay compensation, we present some crucial simulation results. We
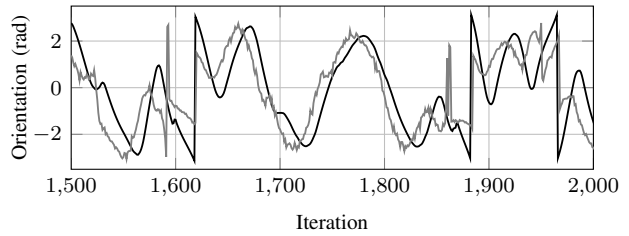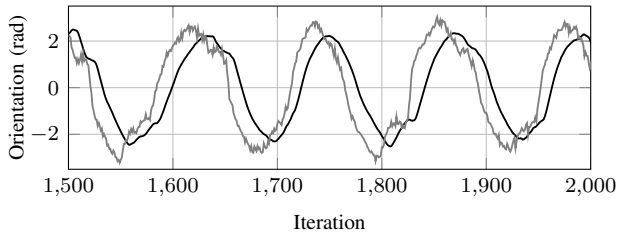
Fig. 6: Simulation result depicting the tracking of the desired velocity angle γ_{ref} (black line) by the resulting velocity angle γ (gray line). A delay of $\tau = 180ms = 6T_{\mathrm{s}}$ exists with line length $L = 25m$. Controller with delay compensation (left) and without delay compensation (right).
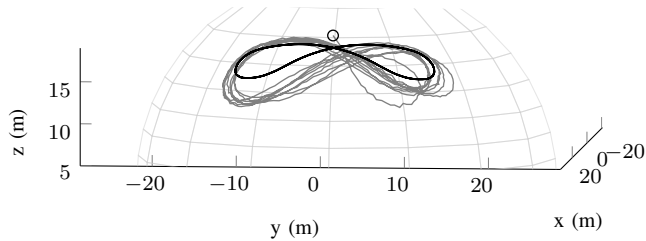

Fig. 7: Experiment: $K_{\mathrm{P}} = 2$, $m = 0.2$, mean velocity = 16.34, kite: HQ Access $3.5m^2$. The black line depicts the desired path $\mathbf{c}_{\mathrm{d}}$, while the gray line depicts the trajectory of the kite, with the circle representing its initial position.
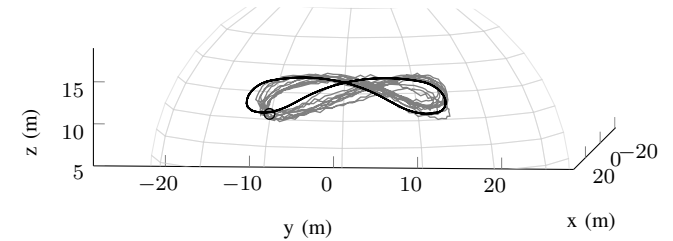
emulate the model inaccuracies of a soft kite, measurement noise and wind turbulence with a realistic simulator (this is a scaled-down version of the simulator presented in [18]).

Figure 6 (left) presents the tracking of γ_{ref} when the delay compensation module is active. The algorithm manages to track closely and repeatedly the desired path. The time lag between γ and γ_{ref} is consistent with the phase shift in the theoretical delay-free closed loop system pertaining to the velocity angle alone. In contrast, in figure 6 (right), where no prediction is performed, tracking is poor and plagued with overshoots. This happens for two reasons: Firstly, the proportional controller has a delayed setpoint γ_{des}, resulting in poor performance. Secondly, the input is delayed, resulting in the overshoot typical of a delayed system governed by a proportional controller. On top of that, given that the algorithm is not able to closely follow the desired path, the choice of the VTP is also altered, resulting in a more erratic γ_{ref} signal.

E. Experimental Results

For the experimental results in this section, we manually executed the launching and landing maneuvers. The focus is on autonomous path-following performance in crosswind conditions.

Figures 7, 8 and 9 depict trajectories generated with the algorithm for three different desired paths. The wind speed during the experiments was approximately 6-8 m/s. We observe that the algorithm achieves repeatable results that are close to the desired path. Moreover, the algorithm is able to perform well under different wind speeds, with different kites, without being overly sensitive to the parameters. A video of the experiments can be found online [19].

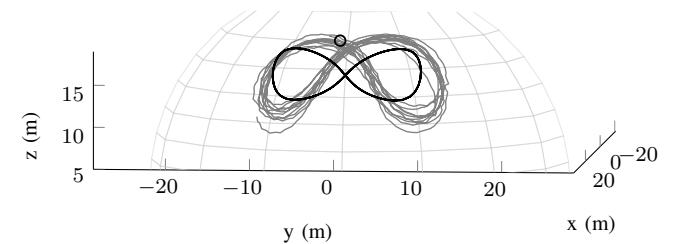In order to better understand the results of the algorithm,

Fig. 8: Experiment: $K_{\mathrm{P}} = 3$, $m = 0.15$, mean velocity = 11.28, kite: Flysurfer Viron $2.5m^2$.

Fig. 9: Experiment: $K_{\mathrm{P}} = 3$, $m = 0.2$, mean velocity = 20.38, kite: HQ Access $3.5m^2$.

a more detailed analysis on a part of the experiment shown in Figure 8 is presented. Figure 10 presents the part under analysis, along with markers that are used to correlate it with the next figures. The average velocity of the kite for the particular experiment was 18.1 m/s.

Figure 11 depicts the relationship between $\bar{\dot{\psi}}$ (lowpass filtered) and $\bar{g}u_{k-N_{\mathrm{d}}}$, which is reasonably accurate. Although for the prediction of ψ, the adaptive filter from subsection III-A is used, the relationship between the $\bar{g}u(k - N_{\mathrm{d}})$ and $\bar{\dot{\psi}}$ is still important, as it is used by Equation (16) in the Guidance Strategy.
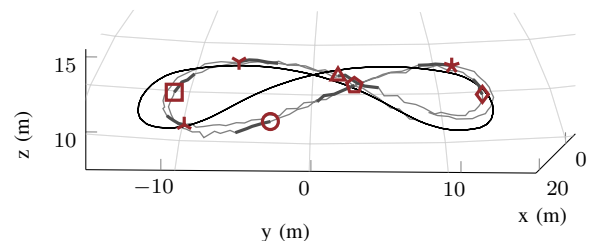

Fig. 10: Experiment: Part of Fig. 8. The black line depicts the desired path $\mathbf{c}_{\mathrm{d}}$, while the gray thin line depicts the trajectory of the kite. Markers are used to correlate with figures 11-15. Thick gray line parts depict the evolution of the trajectory shortly after each marker.
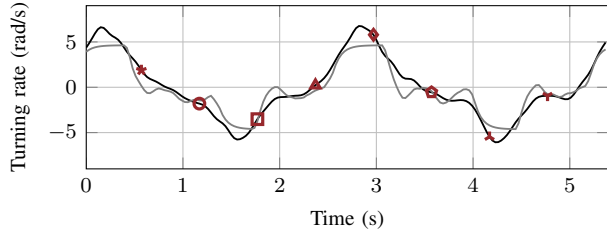
Fig. 11: Comparison between the (filtered) corrected rate of change of the orientation $\dot{\bar{\psi}}$ (black line) and the delayed scaled input $\bar{g}u_{k-N_\mathrm{d}}$ (gray line). Red markers are used to correlate with Fig. 10
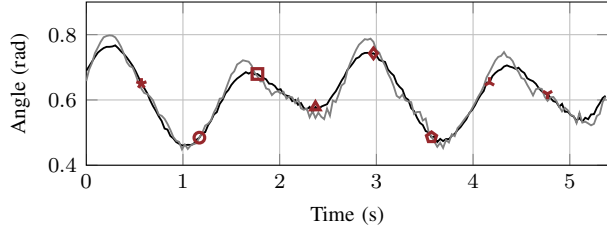

Fig. 12: Prediction of θ (gray) versus (filtered) future values (black).

In figures 12, 13 and 14, we can observe the future (lowpass filtered) values, as well as predictions for θ, ϕ and γ respectively, which are quite accurate.

Finally, figure 15 presents the arc-length parameter of the VTP. Discontinuities occur when a loop is being completed (the length of the desired path is $l_\mathrm{d} \approx 2.18$). The evolution of the VTP is smooth, while there exist points where the VTP remains constant. This happens during the turns of the trajectory and stems from the constraint that the arc-length parameter of the VTP is an increasing function of time.

V. CONCLUSION

As it can be observed from the experimental results of the previous section, the proposed algorithm robustly follows a desired path in realistic wind conditions, with different kites.
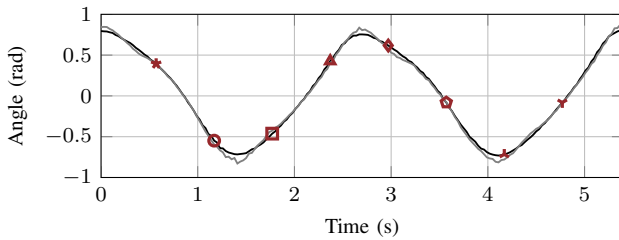

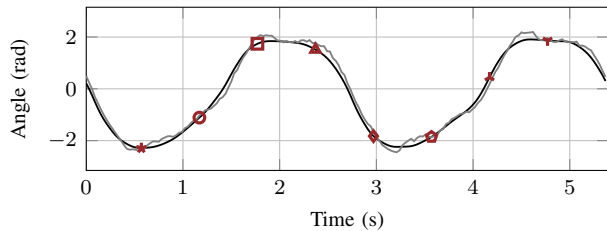Fig. 13: Prediction of ϕ (gray) versus (filtered) future values (black).

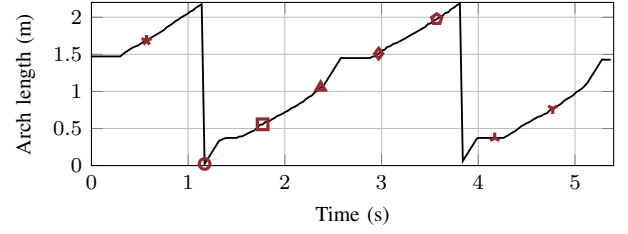Fig. 14: Prediction of γ (gray) versus (filtered) future values (black).

Fig. 15: Arch length parameter of the VTP.

The algorithm was able to run in different environmental conditions, without being overly sensitive to the chosen control parameters.

Moreover, the characteristics of the desired path can be used to reduce the tuning of the parameters. Given the linear dependence of κ_g on u (Eq. (16)), a space varying gain $K_\mathrm{P}(t)$ can be introduced, which will impose more aggressive behavior in the turns, and more moderate behavior in the straight parts, thus avoiding overshoots and improving performance. The maximum desired κ_g can be used to determine m and δ. Finally, statistical characteristics of the measurements can be used to determine $v_{\max}, v_{\min}$. These ideas are under development and the preliminary results are promising.

For a full pumping cycle AWE system, a variable tether length L should be considered [20]. Although most of the controller remains unaffected, the dynamics used in the prediction should be adapted accordingly.

Finally, building on a robust and repeatable control algorithm, a higher level optimization layer can be used to determine the optimal desired path in real time. Research is ongoing in this direction [21].

ACKNOWLEDGMENT

APPENDIX

We define the inertial right handed coordinate system I, consisting of the basis vectors $\mathbf{e}_\mathrm{x}$, $\mathbf{e}_\mathrm{y}$, $\mathbf{e}_\mathrm{z}$. Vector $\mathbf{e}_\mathrm{z}$ points upwards with respect to gravity (azimuth orientation), while the position $\mathbf{p}$ of a particle is defined by the spherical coordinates ϕ and θ. Angle θ is defined as the signed angle between $\mathbf{e}_\mathrm{x}$ and $\mathbf{p}$, while ϕ is defined as the signed angle between $\mathbf{e}_\mathrm{z}$ and the projection $\mathbf{p}_\mathrm{pr}$ of $\mathbf{p}$ to the yz plane.

Using the compact notation $s_\theta := \sin\theta, c_\theta := \cos\theta$ the position $\mathbf{p}$ of a particle in space can be defined as:

$$_\mathrm{I}\mathbf{p} = r \begin{bmatrix} c_\theta \\ s_\theta s_\phi \\ s_\theta c_\phi \end{bmatrix} \tag{18}$$

The local orthogonal unit vectors $\mathbf{e}_\mathrm{r}$ $\mathbf{e}_\theta$, $\mathbf{e}_\phi$ of I, in the directions of increasing $-r$, θ, ϕ, define the local coordinate
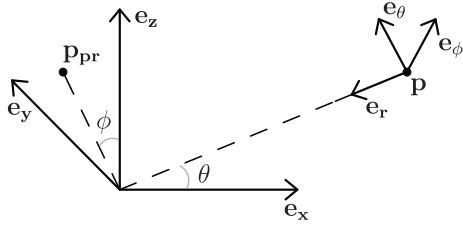
Fig. 16: Coordinate systems I and M.

system M and are defined as:

$$_{\mathrm{I}}\mathbf{e}_r = \begin{bmatrix} -c_\theta \\ -s_\theta s_\phi \\ -s_\theta c_\phi \end{bmatrix}, \; _{\mathrm{I}}\mathbf{e}_\theta = \begin{bmatrix} -s_\theta \\ c_\theta s_\phi \\ c_\theta s_\theta \end{bmatrix}, \; _{\mathrm{I}}\mathbf{e}_\phi = \begin{bmatrix} 0 \\ -s_\theta c_\phi \\ -s_\theta s_\phi \end{bmatrix} \quad (19)$$

For curves lying on a unit sphere the following relationship holds between curvature κ and geodesic curvature κ_{g} [16]:

$$\kappa^2 = 1 + \kappa_{\mathrm{g}}^2 \quad (20)$$

The geodesic curvature represents how much a curve deviates from a geodesic. A geodesic is a generalization of a straight line to a surface. In a sphere, geodesics are circles with the biggest possible radius. The geodesic distance d_{g} between two points $\mathbf{x}_1, \mathbf{x}_2$ on a sphere of radius L can be calculated as:

$$d_{\mathrm{g}}(\mathbf{x}_1, \mathbf{x}_2) = L \arccos\left(\frac{\mathbf{x}_1}{L} \cdot \frac{\mathbf{x}_2}{L}\right). \quad (21)$$

Calculating κ for curves produced by the dynamics (1a)-(1c) with $\theta \in (0, \pi/4)$, $E \gg \tan\theta$ gives:

$$\kappa = \frac{|_{\mathrm{M}}\dot{\mathbf{p}} \times {}_{\mathrm{M}}\ddot{\mathbf{p}}|}{|_{\mathrm{M}}\dot{\mathbf{p}}|^3} = \left| \begin{bmatrix} \dot\theta \\ \dot\phi s_\theta \\ 0 \end{bmatrix} \times \begin{bmatrix} \ddot\theta - \dot\phi^2 s_\theta c_\theta \\ \ddot\phi s_\theta + 2\dot\theta\dot\phi c_\theta \\ \dot\theta^2 + \dot\phi^2 s\theta^2 \end{bmatrix} \right| / \left| \begin{bmatrix} \dot\theta \\ \dot\phi s_\theta \\ 0 \end{bmatrix} \right|^3$$

$$= \sqrt{1 + (Lgu)^2}$$

$$(22)$$

Comparing Eq. (20) and (22) gives $|\kappa_{\mathrm{g}}| = |Lgu|$. The signed equation can be obtained if we calculate directly κ_{g} [16], which requires significantly more calculations.

In reality $E \in (3, 6)$. For a finite glide ratio κ_{g} depends on u, θ, ψ, L, g. Unfortunately, the equations become rather complex when we take into account the term $\tan\theta/E$ in the calculation of the geodesic curvature. Instead, we chose to calculate numerically the accuracy of equation (16). As we can observe from equation (23), equation (16) is sufficiently accurate both for small and large inputs:

$$\max_{u,\theta,\psi} \left| \frac{\kappa_{\mathrm{g}} - Lgu}{Lgu_{\max}} \right| < 4.5\%, \; u \in [0, u_{\max}]$$

$$\max_{u,\theta,\psi} \left| \frac{\kappa_{\mathrm{g}} - Lgu}{Lg\frac{u_{\max}}{10}} \right| < 15\%, \; u \in \left[0, \frac{u_{\max}}{10}\right]$$

for $\theta \in \left(0, \frac{\pi}{4}\right], \psi \in [-\pi, \pi), E = 4, g = 0.5, u_{\max} = 0.4.$

$$(23)$$

REFERENCES

[1] L. Fagiano and M. Milanese, "Airborne wind energy: An overview," in *American Control Conference (ACC), 2012*, June 2012, pp. 3132–3143.

[2] U. Ahrens, M. Diehl, and R. Schmehl, Eds., *Airborne Wind Energy*. Berlin: Springer, 2013.

[3] C. L. Archer and K. Caldeira, "Global assessment of high-altitude wind power," *Energies*, vol. 2, no. 2, pp. 307–319, 2009.

[4] R. Ruiterkamp and S. Sieberling, "Description and preliminary test results of a six degrees of freedom rigid wing pumping system," in *Airborne Wind Energy*, U. Ahrens, M. Diehl, and R. Schmehl, Eds. Berlin: Springer, 2013, pp. 443–458.

[5] M. Erhard and H. Strauch, "Control of towing kites for seagoing vessels," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 5, pp. 1629–1640, Sept 2013.

[6] 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*, vol. 17, no. 17, pp. 1590–1599, 2007.

[7] M. Diehl, "Real time optimization for large scale nonlinear processes," Ph.D. dissertation, Ruprecht-Karls-Universität Heidelberg, 2001.

[8] S. Gros, M. Zanon, and M. Diehl, "Control of airborne wind energy systems based on nonlinear model predictive control amp; moving horizon estimation," in *Control Conference (ECC), 2013 European*, July 2013, pp. 1017–1022.

[9] J. Baayen and W. Ockels, "Tracking control with adaption of kites," *Control Theory Applications, IET*, vol. 6, no. 2, pp. 182–191, January 2012.

[10] 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," pp. 1–1, 2013.

[11] C. Jehle and R. Schmehl, "Applied tracking control for kite power systems," *Journal of Guidance, Control, and Dynamics*, pp. 1–12, 2014/04/30 2014.

[12] P. Sujit, S. Saripalli, and J. Borges Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *Control Systems, IEEE*, vol. 34, no. 1, pp. 42–59, Feb 2014.

[13] J. Breukels, "An engineering methodology for kite design," Ph.D. dissertation, Delft University of Technology, 2011.

[14] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, 4th ed. Prentice Hall, 2006.

[15] S. Park, J. Deyst, and J. P. How, "Performance and lyapunov stability of a nonlinear path following guidance method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2015/03/16 2007. [Online]. Available: http://dx.doi.org/10.2514/1.28957

[16] M. Spivak, *A Comprehensive Introduction to Differential Geometry*, 3rd ed., ser. A Comprehensive Introduction to Differential Geometry. Publish or Perish, 1999, no. 3.

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

[18] S. Costello, G. François, and D. Bonvin, "Kite Control - A Benchmark Problem for Advanced Control and Dynamic Optimization." *Submitted to Journal of Optimal Control Application and Methods.*, 2015. http://infoscience.epfl.ch/record/201679.

[19] N. Rontsis and S. Costello, "Video: Autonomous flight of the small scale awe prototype of epfl." http://youtu.be/LDpPJ8W-WOE, 2014, available online.

[20] A. Zgraggen, L. Fagiano, and M. Morari, "On modeling and control of the retraction phase for airborne wind energy systems," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 5686–5691.

[21] S. Costello, G. François, and D. Bonvin, "Real-time optimization for kites," in *Proceedings of the IFAC Workshop on Periodic Control Systems, Caen, France*, vol. 3, no. 5, 2013.