Real-Time Optimizing Control of an Experimental Crosswind Power Kite

Sean Costello, Grégory François, and Dominique Bonvin

Abstract—The contribution of this article is to propose and experimentally validate an optimizing control strategy for power kites flying crosswind. The control strategy provides both path control (stability) and path optimization (efficiency). The path-following part of the controller is capable of robustly following a reference path, despite significant time delays, using position measurements only. The path-optimization part adjusts the reference path in order to maximize line tension. It uses a real-time optimization algorithm that combines off-line modeling knowledge and on-line measurements. The algorithm has been tested comprehensively on a small-scale prototype, and this article focuses on experimental results.

Index Terms—Path-following control, real-time optimization, airborne wind energy.

I. INTRODUCTION

WIND is one of the most promising renewable energy sources, and kite power is among the most promising emerging wind power technologies. A global study based on experimental data estimated that the world’s energy demand could be entirely satisfied using conventional wind turbines installed on only 2% of the world’s land area [1]. However, in most locations the cost of wind energy is still significantly higher than that of energy produced from fossil sources, and wind power only accounts for about 2% of global electricity production. For wind power to be truly competitive, it needs to become significantly cheaper, more consistent and more efficient. “Airborne Wind Energy” using kites is a promising wind power concept that is radically different from conventional wind turbines. Kites are wings, ranging from flexible para-glider type designs to rigid composite aircraft wings, attached to the ground by a flexible tether. In addition to having low material costs, they can exploit the fact that wind strength and regularity increases with altitude [2], [3], [4]. Powerful kites already have wide application in sports: kite-propelled craft regularly break the world speed-sailing record. Now very large kites are being used to propel cargo ships [5], and many prototypes for electricity production are already in existence. An overview of this development can be found in [6], [7].

A number of significant technical barriers are being actively tackled in order for kite power to become commercially viable. The control of kites is one of the most fundamental challenges. A dynamically flying power kite is a fast, unstable system influenced by unpredictable wind disturbances, and usually only noisy and incomplete measurements are available. It is a testament to the difficulty of stabilizing a kite during dynamic flight that the first successful account of experimental kite control was published in 2013 [5], 33 years after research on kite power began [8]. Typically, in order to extract maximum power from the wind, a power kite is flown almost perpendicular to the wind, similarly to the blades of a wind turbine, reaching speeds many times that of the wind itself (easily in excess of 150 km/h). This is known as crosswind flight, and is the focus of this paper. Note that there are other useful modes of flight, such as static flight at the zenith, which are not treated here. During crosswind flight, if the kite is not constantly steered, it will crash in a matter of seconds. Hence, an “autopilot” must keep the kite flying in a wide variety of wind conditions, providing stability. What is more, unlike the blades of a wind turbine which must move in a circle, a kite can follow many different flight paths. The path the kite flies determines how much power is produced. Hence, in addition to keeping the kite from crashing, the autopilot must ensure the kite follows a path that is efficient for power production.

The field of kite control is young, yet varied. In theory, state-of-the-art Nonlinear Model Predictive Control (NMPC) is the perfect solution [9], [10], [11], given the complex nature of the control problem, the presence of operational constraints, and the necessity to optimize the kite’s flight path. Unfortunately, this has yet to become a practical reality, mainly due to the inaccuracy of existing kite models, as NMPC relies on the quality of the model at hand. Many power kites are of the flexible kind, and accurate models [12], [13] are very complex, generally unsuitable for NMPC. We note, however, that with recent advances in MPC [14], this situation will hopefully change. There are, on the other hand, several accounts of experimentally validated geometric control laws. For example, [15] observed that a simple control scheme should aspire to control the kite’s direction of motion, referred to as the velocity angle, which will also be used extensively in this paper. They combined an online system-identification algorithm with a Lyapunov-based control law. The control law attempts to choose the kite’s velocity angle such that it smoothly attains the prescribed target trajectory. An additional contribution was to elegantly exploit the concept of geodesic curvature to simplify the problem of tracking on a sphere. Also, [5] developed a simple, robust cascade controller for kites, which was tested by years of sea trials on large vessels. Essentially, a low-level proportional controller regulates the
kite’s orientation (i.e. the direction the kite is pointing), while a higher-level guidance controller chooses the bang-bang reference orientation signal, based on the kite’s current position. This results in a horizontal figure-of-eight pattern, which is generally considered to be the most efficient type of path for extracting energy from the wind. The resulting controller has only a few tuning parameters, however the effect of these parameters on the kite’s trajectory is difficult to determine a priori. A similar cascade-control strategy was proposed and experimentally validated on a small prototype by [16]. The primary controlled variable was the kite’s velocity angle in this case, which was again regulated by a simple low-level linear controller. The guidance strategy alternately directs the kite towards one of two points, producing the classic figure-of-eight pattern. The tuning parameters in this case can be used to choose the height, width and inclination of the figure-of-eight in an intuitive manner. The same authors extended this control law to also handle the retraction phase for a pumping-cycle generator, and successfully implemented the algorithm on a power-producing prototype [17]. Building on the work by [15], [18] proposed a more advanced path-following controller using a nonlinear guidance-law and successfully tested it on a 20-kW pumping-cycle prototype. Feedback linearization is used to design a low-level velocity-angle controller. The guidance law aims to minimize the cross-track error, taking into account the kite’s current velocity angle, the path’s direction, and the curvature of the path. Successful implementation of a path-following controller is also reported in [19], this time via waypoint tracking for rigid wings.

While several control solutions for kites now exist, and the most advanced of these are even capable of tracking relatively arbitrary paths, the path-planning problem during crosswind flight is still an open issue. Intelligent path planning is important because, although the kite is free to follow almost any flight path, it is the flight path that directly determines the aerodynamic force the kite experiences, and hence the power generated. Experimental studies have confirmed that the path taken by the kite significantly affects the power it can generate [20]. The path-planning problem results in an interesting optimal control problem that has been studied by a number of authors [9], [21], [22], [23], [24], [25]. However, despite promising recent work [26], there remains a gap between the theory and the application. Recently, more detailed models have been employed [27], particularly for rigid wings for which modeling is easier. Flexible power kites are very popular, and for these kites modeling is still quite approximate. In addition, effects such as wind gradients vary from location to location and cannot be known in advance. Hence, with the tools that are currently available, it remains questionable whether a purely model-based approach can calculate optimal paths for a real flexible kite. This is probably the reason why the most prominent available experimental study uses an experimental approach to tune the path the kite follows; [20], [28] proposed an algorithm that adjusts the height and lateral position of the kite’s path in real time, using experimental data only. These two parameters are then optimized online using a gradient-search algorithm. Since the use of offline modeling to design the real-time optimization algorithm, and online experimental data to perform the optimization, has yielded promising results, this paper will develop a similar approach.

The contribution of this paper is to propose a combined control and optimization strategy for power kites flying crosswind. The control strategy provides both path control (stability) and path optimization (efficiency). The entire algorithm is the result of many years of experimental work, and it has been tested comprehensively. This paper contains experimental data only. The path-following part of the controller is capable of robustly following arbitrary paths, despite significant time delays, using position measurements only. The path-optimization part maximizes line tension by combining offline modeling knowledge and online measurements. The online data ensures it is robust to wind or system variations. Finally, we also propose a new, experimentally validated, modeling relationship for power kites, linking the decrease in the kite’s lift-to-drag ratio to the steering deflection.

The paper is structured as follows. Section II describes the experimental setup. Section III develops a low-dimensional dynamic model of the system that is based on existing models from the literature, proposes a novel addition, and validates the model using experimental data. Section IV presents a path-following controller for kites and analyzes its experimental performance. Section V presents the Real-Time Optimization (RTO) algorithm, describes how the effect of noise is mitigated, and illustrates its experimental performance.

II. Experimental Setup

The experimental setups used in this work are small (2.5 and 3.5-m²) kites on a short (35 m), fixed-length line, sensed and actuated from the ground by a mechanized station. This small-scale setup has the advantage of providing a relatively controlled environment, while retaining many of the properties of the much larger kites used in commercial settings, which typically use line lengths of 50-500 m and kite sizes of 10-300 m². Firstly, short lines experience negligible line drag during crosswind flight. This allows the kite’s position to be quite precisely measured (to within several cm) using line angles, and the attitude and the steering deflection to be precisely controlled. Secondly, short lines keep the kite close to the ground, that is, where the wind speed and direction are measured. Hence, the small-scale system provides a controlled environment for validating modeling hypotheses and control strategies.

A. Physical System

The kites employed, one of which is shown in Figure 1, have the characteristics of standard power kites. Two commercial power kites were used: A 2.5-m² Flysurfer Viron and a 3.5-m² HQ Apex. Both are three-line² kites; one front line takes

1The Reynolds number of both a 3-m² kite and a scaled-up 300-m² kite in crosswind flight ensures turbulent flow [25], thus there are unlikely to be significant differences between their aerodynamic properties, and hence their behavior.

2This type of kite is often termed a 4-line kite also, as there are 4 attachment points on the kite. However, the two front lines join together, and there are only 3 attachment points on the control bar.
about 90% of the force generated by the kite, the two lightly-tensioned rear lines allow the kite to be maneuvered. There are two degrees of freedom to operate the kite: (i) adjusting the difference between the lengths of the rear lines allows the kite to be steered left or right, and (ii) adjusting the length of the front line allows the kite to be accelerated or decelerated by changing its angle-of-attack relative to the on-rushing air. In this case, only the steering degree of freedom is adjusted automatically. The length of the front line is maintained constant throughout each experiment. Hence, in this work, the kite can be steered but neither its speed, nor the line tension it produces, can be directly controlled.

The geometry of the ground station, shown in Figure 2, has an important impact on the experiments. The lines coming from the kite are led through three small eyes placed closely together. This ensures that, as the kite moves around, and hence the angle of the lines changes, the relative lengths of the lines will not vary. The front line is attached to a load cell that measures the line tension. The angle of the front line between the station and the kite is measured by a 1-m long light carbon-fiber rod with a small ring at the end, through which the front line passes. The angle of the rod, from which the line angle can be inferred, is measured by rotary encoders. The rear (steering) lines are wound in opposite directions around a reel, which is turned by a responsive and powerful servomotor. Rotating the reel shortens one line, while lengthening the other, achieving a steering effect. A high-precision ultrasonic anemometer mounted on a 3-m pole measures the wind speed and direction. The control algorithm runs in real time on a laptop. Hence, accurate measurements of the kite’s position, the front-line tension and the wind speed and direction are available at the ground station, where the steering input is manipulated.

B. Testing Conditions

Wind speed is the most obvious testing condition that influences the kite’s behavior. It affects the speed of the kite, the line tension it produces, and the kite’s turning characteristics. Of equal importance, although less obvious, is the short-term variability of the wind’s speed and direction, i.e. the gustiness of the wind, which is mostly decided by the ruggedness of the local terrain.

Tests were performed in locations with very different levels of gustiness, from (not very gusty) beaches to (very gusty) mountain ridges and narrow valleys. Very gusty conditions severely complicate modeling, but on the other hand, controllers should show robustness to a reasonable level of gustiness. The results presented here were obtained in (medium gusty) flat fields, with no obstructions to windward for at least 500 meters. The average wind-speed was approximately 4 m·s$^{-1}$, which is a gentle breeze.

Nonetheless, it is important to note that the path-following control strategy was successfully validated in a wide variety of conditions, including very strong and gusty winds. In summary, for the sake of conciseness, this paper uses one set of testing conditions that allow for reasonably accurate modeling, while still presenting a realistically difficult control challenge.

III. Modeling

This section focuses on low-state-dimension dynamic models for flexible power kites. Much progress has been made in modeling the dynamics of flexible kites during the last decade [9], [21], [29], [30], [31], [13], [32], [12], [33], [34]. Relatively simple tendency models have between 3 and 10 states, while more complex models aimed at achieving great precision have up to several hundred states. The complex models are constructed afresh for each kite geometry, and are generally used in a simulation environment to validate control strategies. For the purposes of this article, tendency models are more useful, since they incorporate general characteristics that are reproduced by most flexible power kites and, through their simplicity, allow interesting insights into the kite’s dynamic behavior. The two tendency models that are typically used for controller design and path optimization are the well-established point-mass model [9], [16], and that proposed by [5], henceforth referred to as the Erhard Model. Both models are general enough to apply to both ram-air
kites and tube kites. If appropriate parameters are used, both models will predict qualitatively similar behavior. However, the Erhard Model is simpler and more intuitive, as it has only two aerodynamic parameters that can be calculated for a real system using straightforward experiments. Consequently, it has been successfully used to design a control algorithm for very large, commercial kites. Hence, as one of the simplest and best-validated flexible kite models, the Erhard Model is the starting point for understanding the experimental system in this paper. Next, the Erhard Model is described, then a second model based on the kite’s velocity angle is derived, and finally a novel addition to the turning law is derived from experimental data.

A. Erhard Model

The Erhard Model [5] is surprisingly simple due to its choice of inertial, right-hand co-ordinate system that is depicted in Figure 3. Choosing the zenith to be aligned with the wind direction significantly simplifies the dynamic equations, as the kite’s behavior is symmetric about this axis if the effect of gravity is neglected. For flexible kites, which are very light, this is a reasonable assumption during crosswind flight when the aerodynamic forces far outweigh gravity. The kite’s position in Cartesian coordinates is given by:

$$
\mathbf{p} = r \begin{bmatrix} \cos \vartheta \\ \sin \vartheta \sin \varphi \\ \sin \vartheta \cos \varphi \end{bmatrix},
$$

where \( r \) is the (constant) length of the kite’s tether, and \( \vartheta \) and \( \varphi \) are spherical coordinates for the kite’s position, using the \( x \)-axis as the zenith. In this paper, the kite’s position is often represented as a projection onto the \( \{ N, W \} \) plane shown in Figure 3. The plane is defined by the two orthogonal vectors \( \hat{e}_W = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \) and \( \hat{e}_N = \begin{bmatrix} 0 & \sin \vartheta & 0 \cos \vartheta \end{bmatrix}^T \), which are tangent to the sphere upon which the kite can move at the point \( \{ \vartheta, \varphi \} = \{ \vartheta, 0 \} \).

The dynamic equations for the model are:

$$
\dot{\vartheta} = \frac{w_{ap}}{r} \left( \cos \psi - \tan \vartheta \frac{E}{E} \right),
$$

$$
\dot{\varphi} = -\frac{w_{ap}}{r \sin \vartheta} \sin \psi,
$$

$$
\dot{\psi} = w_{ap} g_s \delta + \dot{\varphi} \cos \vartheta,
$$

where \( \psi \) is the kite orientation, \( g_s \) is the turning constant, and \( E \) is the kite’s lift-to-drag ratio. The steering deflection, \( \delta \), is the system’s manipulated variable. \( w_{ap} \) is the magnitude of the apparent wind projected onto the plane that is normal to \( \mathbf{p} \), and is given by:

$$
w_{ap} = w E \cos \vartheta,
$$

where \( w \) is the wind speed at the kite’s current altitude. A number of different wind-shear models exist to describe the variation of wind speed with altitude. One of the most common is the power law [3]:

$$
w = w_{ref} (z/z_{ref})^a,
$$

where \( a \) is the surface friction coefficient, \( w_{ref} \) is the reference wind speed at the reference altitude \( z_{ref} \), and \( z \) is the kite altitude. Finally, the line tension is given by

$$
T = \left( \frac{1}{2} \rho A w^2 \right) (E + 1) \sqrt{E^2 + \cos^2 \vartheta}.
$$

Hence, using only three states representing position and orientation, the Erhard Model describes the kite’s velocity, turning behavior and line tension.

B. Cart Model

The Erhard Model is remarkably straightforward, but it can be cast into an even simpler form that is more useful when sensors are not mounted on the kite. The first two states in this model, the kite’s position, are generally measured on all experimental systems. However, an accurate measurement of the third state, the kite’s orientation, is not always so readily available. This is certainly the case for our experimental setup, which has no gyroscope on the kite. Luckily, while the kite’s orientation is no doubt a useful measurement in any situation, it is not essential for control during crosswind flight, during which the kite is moving rapidly, and the aim is to control the kite’s direction of motion rather than its orientation. We will use an alternative state variable, the velocity angle, which has been used for kite control by [16], [18]. The velocity angle is the angle of the kite’s velocity projected onto the plane that is tangent to the sphere at the kite’s current position. In fact, it will be seen that, during crosswind flight, the kite’s orientation is approximately the same as its velocity angle, with the advantage of the velocity angle being that it can be inferred from a sequence of position measurements. Hence, the resulting model, which will be referred to as the Cart Model, can be used for modeling and control when only position measurements are available.

The key assumption for deriving the following Cart Model is that the kite flies crosswind, which occurs when the lines of the kite do not make a large angle with the wind vector, i.e. when \( \tan \frac{\vartheta}{E} \ll 1 \). Importantly, during crosswind flight, the speed of the kite is always greater than the wind speed. We begin by introducing the velocity angle, as defined by [16]:

$$
\gamma = \tan^{-1} \left( \frac{\dot{\varphi} \sin \vartheta}{\dot{\vartheta}} \right).
$$
Developing the expression for the velocity angle (using Equations III.2 and III.3) gives:

$$\gamma = \tan^{-1} \left( \frac{-\sin \psi}{\cos \psi - \tan \theta \frac{E}{r}} \right). \quad (III.9)$$

As $\frac{\tan \theta}{E} \ll 1$ in crosswind flight, it follows that:

$$\gamma \approx -\psi. \quad (III.10)$$

In other words, we have shown that, not surprisingly, the kite flies crosswind in roughly the same direction as it is pointing. This is in accordance with the low sideslip (or drift) assumption made by [16], [18]. Next, we examine the kite’s speed relative to the tether length (i.e. in m·s$^{-1}$) is:

$$\omega_k := \frac{\|\mathbf{p}\|}{r} = \sqrt{(\dot{\varphi} \sin \theta)^2 + (\dot{\psi})^2}. \quad (III.11)$$

Next, inserting the differential Equations III.2 and III.3 gives:

$$\omega_k = \sqrt{\left( -\frac{w_{ap}}{r} \sin \psi \right)^2 + \left( \frac{w_{ap}}{r} \left( \cos \psi - \frac{\tan \theta}{E} \right) \right)^2}$$

$$= \frac{w_{ap}}{r} \sqrt{(\sin \psi)^2 + \left( \cos \psi - \frac{\tan \theta}{E} \right)^2}. \quad (III.12)$$

Finally, as $\frac{\tan \theta}{E} \ll 1$ in crosswind flight:

$$\omega_k \approx \frac{w_{ap}}{r} = \frac{w}{r} \cos \vartheta. \quad (III.13)$$

Hence, during crosswind flight, the kite’s speed approximately depends only on the position, and not the orientation, of the kite. Using the definition of $\gamma$ from Equation (III.8) and the expression for the kite’s angular velocity from Equation (III.11), the Cart Model for the kite reads:

$$\dot{\vartheta} = \omega_k \cos \gamma, \quad (III.14)$$

$$\dot{\psi} = \frac{-\omega_k}{\sin \gamma} \sin \gamma, \quad (III.15)$$

$$\dot{\gamma} = -\left( \omega_k r g_s \delta + \dot{\vartheta} \cos \vartheta \right), \quad (III.16)$$

$$\omega_k = \frac{w}{r} \cos \vartheta. \quad (III.17)$$

These dynamics are indeed the same than those of a cart driving on a sphere, with affine steering dynamics, whose speed cannot be directly controlled, justifying a posteriori the label of Cart Model. If unlimited steering action is assumed, any path can be achieved, but in reality the steering deflection is bounded. Note that, if $E$ is assumed to be constant (which, as we shall see, is reasonable if the model is to be used to design steering controllers, but not for tether-force optimization), the input required to follow any smooth path $[\vartheta(t), \psi(t)]$ can be easily computed, although it may not satisfy the maximum steering-deflection constraint. This compact model is simple enough that the values of the free parameters can easily be identified experimentally, yet it will nonetheless allow achieving good control performance during crosswind flight.

C. Experimental Characterization of the Kite’s Turning Behavior

At this point, it is important to see how this model compares to actual experimental data. It will be seen that two additional characteristics of the kite’s turning behavior must be accounted for. Let the corrected turning rate be defined as:

$$\dot{\gamma}_c = \dot{\gamma} + \dot{\vartheta} \cos \vartheta. \quad (III.18)$$

Injecting Equation (III.18) into Equation (III.16) leads to the following proportional relationship:

$$-\frac{\dot{\gamma}_c}{\omega_k r} = g_s \delta. \quad (III.19)$$

The constant $g_s$ can be estimated from experimental data, as shown in Figure 4, by finding the value of $g_s$ that results in the two signals overlaying each other. Clearly, $-\frac{\dot{\gamma}_c}{\omega_k r}$ is only approximately proportional to the steering input $\delta$. More important, there is a significant delay between a change in the steering deflection and the resulting change in the turning rate. Note that this delay occurs at the level of the kite and the lines, as all software and actuation delays inherent in the ground station have already been accounted for in Figure 4. Although not well understood from a modeling perspective, the delay is most likely due to the rotational inertia of the kite and the flexibility of both the lines and the kite itself. This significant delay is a very important aspect of the kite’s behavior, and the manner in which it is dealt with will largely affect the control performance.

The second experimentally observable phenomenon is the effect of turning on the line tension. An optimizing control strategy for a kite will generally aim to maximize some function of the line tension, for example the component of the line tension in a particular direction for vehicle traction. The Erhard Model (and hence the Cart Model) predicts that the line tension and the kite’s speed simply depend on the angle between the tether and the wind, $\vartheta$, if $E$ is assumed constant. Although, as we shall see, this assumption is not at all true, it is certainly a reasonable approximation if the model is used for directional control of the kite, where subtle variations in the kite’s speed and the line tension are of secondary importance. However, for optimization, such variations in the line tension are of primarl importance. During experiments, it was observed...
that steering deflections cause a significant reduction in tether tension. This is particularly noticeable for very large steering deflections, which will almost cause the kite to stall, drastically reducing the apparent wind speed, and hence the tether tension. We propose to link steering deflections to a reduction in the lift-to-drag ratio using the following empirical law:

\[ E = E_0 - c\delta^2, \quad \text{(III.20)} \]

where \( c \) is a constant that determines how much the kite’s lift-to-drag ratio is penalized for a steering deflection. The basic idea is that the two mechanisms by which flexible kites steer, banking and introducing twist along the wing, tend to decrease the kite’s effective lift-to-drag ratio, which affects the tether tension. Both of these effects can be modeled as a decrease in \( E \) that depends on the steering deflection \( \delta \). A quadratic dependence was chosen as it best fit the experimental data. To validate this hypothesis, so-called ‘bang-bang’ experiments were carried out, during which the magnitude of the input was maintained constant, while an operator commuted the sign (steering direction) between positive and negative (left and right) to keep the kite flying crosswind. Based on Equation (III.5), the lift-to-drag ratio corresponding to each value of \(|\delta|\) was then estimated as:

\[ E = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} \frac{\bar{w}_p \cos \vartheta}{\bar{w}} dt, \quad \text{(III.21)} \]

where \( \bar{w} \) is the average measured wind speed between \( t_0 \) and \( t_f \). For each value of \(|\delta|\), a time period \( t_f - t_0 \) of at least 1 minute (corresponding to more than 30 turns of the kite) was used. The results, along with the fitted curve for \( E \), are shown in Figure 5. It can be seen that large steering deflections result in a reduction of over 30% in the lift-to-drag ratio, which results in a reduction in line tension of over 50%! Incorporating this steering-penalty law into the model when performing path optimization is key to obtaining meaningful results.

Aerodynamic parameters for the experimental system, estimated during an experiment with a wind speed of 9 m·s\(^{-1}\), are given in Table I. Note that, although we cannot currently model this precisely, the values were observed to vary with wind speed.

### IV. Path-Following Control

Using the Cart Model, the path-following cascade control loop shown in Figure 6 can be constructed. Firstly, the missing state, the velocity angle, is estimated from the kite’s measured position. This is described in Section IV-A. Next, an “adaptive prediction” block advances the measurements forward in time to counteract the inevitable time delays inherent in the system that include the delay introduced by the velocity estimation technique, the computational delay, the delay introduced by filters, the actuator response delay and the aforementioned steering delay of the kite of Section III-C. The adaptive prediction block acts in exactly the opposite manner as a time delay by taking the current position measurement and the estimated velocity angle and integrating an adaptive model of the kite forward in time to predict the future values of these states. The adaptive prediction algorithm has been published in a separate article [35] and will not be detailed here. Note that delay is a clear problem for kite control [18], and other compensation approaches have also recently been proposed [36], [37].

The guidance strategy block uses a geometric algorithm to produce a reference velocity angle such that the kite will follow the reference path \( \{\theta_r(t), \varphi_r(t)\} \). The guidance strategy is described in Section IV-B. This reference path itself is periodically updated by the RTO algorithm to be described in Section V. A low-level proportional controller tracks the velocity-angle setpoint. Section IV-C shows the impact of delay compensation on this control loop during experiments. Finally, results following different paths are presented in Section IV-D.

#### A. Velocity-Angle Estimation

At the sampling instant \( n \), an estimate of the kite’s velocity in Cartesian coordinates, \( \hat{p} \), is obtained by solving the following least-squares problem:

\[ \{\hat{p}, \hat{a}\} = \arg\min_{\{p, a\}} \sum_{j=0}^{2d_d} \|(a - jT_s p) - \hat{p}[n-j]\|^2 \quad \text{(IV.1)} \]

where \( \hat{a} \) is an estimate of the kite’s current position. Solving this problem fits a line to the last \( 2d_d + 1 \) position estimates, and the velocity vector associated with the fit is \( \hat{p} \). This is much simpler than using a Kalman filter, while remaining quite robust. However, this approach introduces a delay of \( d_d \) sampling periods for the estimation of kite’s velocity.

#### B. Guidance Algorithm

This section develops a geometric “vector-field” path-following controller for kites. This type of controller is popular
Next, using Equations (III.14) and (III.15), a relationship

\[
\dot{p} = rC \begin{bmatrix} \dot{\varphi} \\ \dot{\varphi} \sin \vartheta \end{bmatrix}, \quad \text{with} \quad C = \begin{bmatrix} -\sin \vartheta & 0 \\ \cos \vartheta \sin \varphi & \cos \varphi \end{bmatrix}.
\]  

Next, using Equations (III.14) and (III.15), a relationship

\[
b = rTC \begin{bmatrix} \dot{\varphi} \\ \dot{\varphi} \sin \vartheta \end{bmatrix}, \quad \text{with} \quad \begin{bmatrix} \dot{\varphi} \\ \dot{\varphi} \sin \vartheta \end{bmatrix} = \omega_k \begin{bmatrix} \cos \gamma \\ \sin \gamma \end{bmatrix},
\]  

where we recall that \( \omega_k \) is the kite’s speed in rad·s\(^{-1} \). We define the kite’s velocity angle on the \( \{ N, W \} \) plane as:

\[
\zeta = \angle \mathbf{b} = \tan^{-1} \left( \frac{b_W}{b_N} \right). \tag{IV.5}
\]  

The reference path (on the \( \{ N, W \} \) plane) is denoted \( \mathbf{b}_r(l) \), where \( l \) is the path length. The points on the path at which the path’s tangent is perpendicular to the kite position are \( \mathbf{b}_r(l_i) \). The angle of the path at each point is denoted as:

\[
\zeta^i = \angle \frac{\partial \mathbf{b}_r}{\partial l} (l_i), \tag{IV.6}
\]

while the vector pointing from the kite to each point is:

\[
d^i_\perp = \mathbf{b}_r(l_i) - \mathbf{b}. \tag{IV.7}
\]

A desired velocity angle corresponding to each point is obtained with the classic vector-field law [38]:

\[
\zeta^i_d = \zeta^i + \zeta_e \left( \frac{\|d^i_\perp\|}{d_{\text{max}}} \right)^\beta \times \text{sgn} \left( \angle \left( d^i_\perp - \frac{\partial \mathbf{b}_r}{\partial l} (l_i) \right) \right) + \alpha \frac{\partial^2 \mathbf{b}_r}{\partial l^2} (l_i), \tag{IV.8}
\]

where the entry velocity angle \( \zeta_e \) and the coefficient \( \beta > 1 \) are tuning parameters. The final term is a new (at least in the context of the vector-field controller) curvature-compensation term that helps the controller follow a curved path. The curvature of the path indicates the rate of change of the path’s angle (direction). Thanks to the curvature compensation (which can be varied by adjusting \( \alpha \)), the controller anticipates curves in the path. Finally, the reference velocity angle is selected as the \( \zeta^i_d \) that is closest to the kite’s current velocity angle:

\[
\zeta_r = \zeta^i_d, \quad i_r = \arg \min_i \| \zeta - \zeta^i_d \|. \tag{IV.9}
\]

The reference velocity angle in \( \{ N, W \} \) coordinates can be translated back into a reference signal for the velocity-angle control loop by inverting Equation IV.4:

\[
\gamma_r = \tan^{-1} \left( \frac{x_2}{x_1} \right), \quad \text{with} \quad x = (TC)^{-1} \begin{bmatrix} \cos \zeta_r \\ \sin \zeta_r \end{bmatrix}. \tag{IV.10}
\]

C. Velocity-Angle Control

We will show next that for the velocity angle, a simple proportional controller yields good performance, provided delay is compensated for. Firstly, note that the final term in Equation (III.16) is generally relatively small, meaning that
the velocity-angle dynamics are approximately those of an integrator. Hence, a proportional controller can be expected to yield good performance. Moreover, a fixed gain is sufficient as, according to this equation, the responsiveness to the control input is proportional to the kite’s speed. The performance during an experiment with adaptive prediction is shown in Figure 8. The closed-loop response is very good, and it was found that the addition of a feed-forward signal to compensate for the final term in Equation (III.16) did not improve the performance significantly. On the other hand, the performance obtained during an experiment without adaptive prediction is shown in Fig. 9. The oscillations around the reference signal are typical of proportional control applied to a first-order system with significant time delay. In our case, there is a total delay of about 260 ms, during which the kite can travel up to 10 m, with a horizontal width of the entire reference path being potentially limited to 15 m! A very important peculiarity of path-following control can be observed, namely, oscillations of the controlled variable cause oscillations of the reference signal! This is because the reference signal depends on the kite’s position. These variations in the reference signal are likely to cause further oscillations in the controlled variable, as the poorly performing controller tries to track them. The overall effect is poor path following and, for this reason, it was found that the response of the velocity control loop must absolutely not overshoot or oscillate. Hence, it was best to choose a conservative value for the proportional gain.

D. Results

The control strategy was chosen by comparing a number of different approaches and by building upon the work of other researchers [5], [16], [18]. We believe the performance is quite good for experimental conditions. Most importantly, the controller shows robustness. The tracking of different reference paths is shown in Figures 10, 11 and 12. Each of these graphs represents 10 minutes of experimental data, during which the kite flies roughly 150 loops. It can be seen that the controller is very consistent, that is, the path followed by the kite is very similar from one loop to the next. The path followed by the kite is not exactly the reference path, and there is a small but consistent offset between the two.

This is because the vector-field control law is heuristic. It is essentially a path-following equivalent of a proportional-derivative (PD) controller. Hence, while it can be tuned to provide good and robust performance, it will not provide exact path-following. In most cases, it is preferred that the controller be consistent and robust to wind variations, and a small path-following error is not a major issue. However, more exact tracking could probably be achieved using run-to-run control techniques.
V. OPTIMIZATION OF THE KITE’S EFFICIENCY

The previous section was about ensuring that the kite’s autonomous flight be predictable and robust to wind variations. It is also desirable that the kite should fly as efficiently as possible, which is the focus of this section. In this work the average line tension, that is how much the kite "pulls", is used as a measure of efficiency. This is closely related to the efficiency measures for electricity-generating kites or for ship-towing kites. In theory, once a model of the kite and the wind is available, numerical optimization can be used to compute the optimal reference path that yields maximal efficiency. However, this approach is difficult in practice, as the models at our disposal are relatively inaccurate. In addition, due to changing wind conditions, the optimal reference path may vary over time. The solution adopted here is to implement a real-time optimization (RTO) layer, which adjusts the reference path in order to maximize efficiency. The RTO algorithm uses both feedback data from the kite and numerical optimization. This feedback component ensures robustness to modeling errors, control errors and disturbances.

A. RTO using Modifier Adaptation (MA)

A RTO method called Modifier Adaptation (MA) is used. This algorithm was initially developed to address the negative effect of modeling inaccuracies on the results of numerical optimization for multivariable industrial processes [39]. The key philosophy behind MA is to repeatedly solve the model-based optimization problem by adding measurement-based modifications to the cost and constraint functions so as to reject the detrimental effect of disturbances and plant-model mismatch. The main advantage of MA lies in its ability to converge to the “true” plant optimum, provided the model satisfies a certain number of fairly relaxed adequacy conditions [40], [41]. Details regarding the theoretical properties, the initialization and the tuning of the specific algorithm used for kite optimization can be found in [42], [43]. A simplified outline of the algorithm employed is given next.

Algorithm V.1: Modifier Adaptation [42]

**Initialization:** Initialize the modifier terms $\epsilon_0 = 0, \lambda_0^j = 0, \forall j \in [1, \ldots, n_g]$, where $\epsilon$ is the so-called $(n_g \times 1)$-dimensional vector of zeroth-order modifier terms for the constraints, $\lambda$ is the $(n_u \times 1)$-dimensional vector of first-order modifier terms for the cost function and $\lambda_k^j$ is the $(n_u \times 1)$-dimensional vector of zeroth-order modifier terms for the $j^{th}$ constraint, $\forall j \in [1, \ldots, n_g]$. Initialize $u_0$ with a conservative input that is unlikely to violate the plant constraints. Select the filter matrices $K_i$ of dimension $(n_g \times n_g), K^\phi$ and $K^{\epsilon}$ of dimension $(n_u \times n_u)$, where $n_u$ is the number of inputs, as, typically, diagonal matrices with eigenvalues in the interval $(0,1]$. Choose the maximum input step magnitude $\Delta u_{\text{max}}$, and the exploration reward factor $c$. Initialize the exploration direction $\delta u_0 = 0$.

1. Solve the modified model-based optimization problem

   $u_k := \underset{u}{\text{argmin}} \phi_{m,k-1}(u)$

   subject to $g_{m,k-1}(u) \leq 0,$

   $\|u - u_{k-1}\| \leq \Delta u_{\text{max}},$  \hspace{1cm} (V.1)

   where the modified cost and constraints are given by

   $\phi_{m,k-1}(u) := \phi(u) + (\lambda_{k-1}^g)^T(u - u_{k-1}) - R(u),$  \hspace{1cm} (V.2)

   $g_{j,k-1}(u) := g_j^1(u) + \epsilon_p \|u - u_{k-1}\|,$  \hspace{1cm} (V.3)

   where $\phi_{m,k-1}(u)$ is the $m^{th}$ modified constraint, $g_j$ is the $j^{th}$ modelled constraint and $\epsilon$ is the $j^{th}$ element of the vector of zeroth-order modifier terms $\epsilon$.

2. Apply the input $u_k$ to the plant and measure the resulting cost and constraint values $\phi_p(u_k)$ and $g_{E,k}$.

3. Incorporate the new measurements into the running estimates of the plant cost gradient, $\nabla \phi_{E,k}$, and the plant constraint gradients, $\nabla g_{E,k}$, using a weighted rank-1 (Broyden) update:

   $\nabla \phi_{E,k} = \nabla \phi_{E,k-1} + \kappa_k (\nabla \delta u \phi_{E} - \nabla \phi_{E,k-1} \delta u) \delta u^T,$  \hspace{1cm} (V.5)

   where

   $\delta u = \frac{u_k - u_{k-1}}{\|u_k - u_{k-1}\|}$  \hspace{1cm} (V.6)

   $\nabla \delta u \phi_{E} = \frac{\phi_p(u_k) - \phi_p(u_{k-1})}{\|u_k - u_{k-1}\|}$  \hspace{1cm} (V.7)

   and the optimal weighting factor $\kappa_k$ is computed according to a formula that minimizes the variance of $\nabla \delta u \phi_{E}$ (see [43] for details). The input direction that maximizes this variance, $\delta u_{E}$, is used as the exploration direction for the next iteration.

$\delta u_{E}$ is computed as follows:

   $\delta u_{E} := \underset{u}{\text{argmin}} \|\nabla \delta u \phi_{E}\|_2 \delta u^T,$  \hspace{1cm} (V.8)

   subject to $\|u - u_{k-1}\| \leq \Delta u_{\text{max}}.$  \hspace{1cm} (V.9)

3The procedure is identical for each of the constraint gradient estimates. When there are constraints, the variance of the Lagrangian function can be used to calculate the exploration direction [43].
4) Update the modifier terms using the following first-order filter equations:

\[
\begin{align*}
\epsilon_k^\phi & := (I_{n_u} - K^\phi)\epsilon_{k-1}^\phi + K^\phi (\dot{g}_{p,k} - g(u_k)) \quad \text{(V.8)} \\
\lambda_k^\phi & := (I_{n_u} - K^\phi)\lambda_{k-1}^\phi + K^\phi \left( \nabla \phi_{E,k} - \frac{\partial \phi}{\partial u}(u_k) \right)^T \quad \text{(V.9)} \\
\lambda_k^\delta & := (I_{n_u} - K^\delta)\lambda_{k-1}^\delta + K^\delta \left( \nabla g_{E,j,k} - \frac{\partial g_j}{\partial u}(u_k) \right)^T \quad \forall j \in [1, \ldots, n_g]. \quad \text{(V.10)}
\end{align*}
\]

end

In Step 1, numerical optimization of a model that represents the cost function \( \phi(u) \) and the constraint functions \( g(u) \) is used, with \( u \) the vector of decision variables. At each iteration, these functions are modified using measurements. The modification terms, which are defined in Step 4, correct the values of the constraint functions and the gradients of the cost and constraint functions. A running estimate of the experimental gradient is implemented in Step 3, after having incorporated the most recent measurements.

The MA algorithm requires a certain amount of tuning, and this mainly centers around ensuring accurate gradient estimates. There is a well-known trade-off between rapidly reaching the plant optimal solution and ensuring accurate gradients estimates, commonly referred to as the exploitation-exploration trade-off. This trade-off is tuned using the exploitation-reward term, \( c \), which rewards exploration in the input direction for which the gradient estimate is most inaccurate.

B. MA for Kite Path Planning

MA can be set up to achieve different objectives via the choice of the cost and constraint functions. In this work, the aim is to maximize the average line tension, and so the measured cost is defined as:

\[
\phi_p(u) := -\bar{T}, \quad \text{(V.11)}
\]

where \( \bar{T} \) is the average line tension measured during \( N_{\text{avg}} \) cycles following the (periodic) reference path. The only operational constraint is that the kite must fly above a certain minimum altitude, \( z_{\text{min}} \), and so the measured plant constraint is defined as:

\[
g_p := z_{\text{min}} - z_L, \quad \text{(V.12)}
\]

where \( z_L \) is the lowest altitude attained by the kite during \( N_{\text{avg}} \) cycles. The decision variables \( u \) must also be defined in terms of both nature and number. This is a critical choice since too many degrees of freedom will increase the exploration load, while the primary objective is to optimize the performance. In this work, the inputs are chosen as the parameters that determine the height and the curvature of the reference path. The reference path is parametrized by \( u = [u_1 \ u_2] \) in the following manner:

\[
\begin{bmatrix}
\vartheta_T(l) \\
\varphi_T(l)
\end{bmatrix} = \begin{bmatrix}
\vartheta^*_T(l, \vartheta_0) \\
\varphi^*_T(l, \vartheta_0)
\end{bmatrix} + u_1 \begin{bmatrix}
\Delta \vartheta_H(l) \\
\Delta \varphi_H(l)
\end{bmatrix} + u_2 \begin{bmatrix}
\Delta \vartheta_C(l) \\
\Delta \varphi_C(l)
\end{bmatrix}, \quad \text{(V.13)}
\]

where \( \{\vartheta^*_T(l, \vartheta_0), \varphi^*_T(l, \vartheta_0)\} \) is the nominal optimal reference path calculated using the model, and \( \Delta \vartheta_H(l), \Delta \varphi_H(l) \) and \( \Delta \vartheta_C(l), \Delta \varphi_C(l) \) are variations that cause raising and widening of the reference path, respectively. The influence of these variations is shown in Figures 13 and 14. This particular choice of \( u \) is based on a mathematical sensitivity analysis with respect to the model uncertainty, the full details of which are given in [42]. Note that allowing the RTO layer to adjust the reference path this way can be justified as follows:

- simultaneous modification of these two variables allows the generation of many different “figures-of-eight” paths,
- the height of the trajectory can be adjusted to suit the wind shear, and
- the curvature (or the width) of the trajectory can be adapted to suit the kite’s turning behavior.

Finally, the model-based cost function \( \phi(u) \) is deduced from the Cart Model developed in Section III-B. This function returns the line tension that is theoretically obtained if the reference path defined by \( u \) is exactly followed. The RTO block diagram is given in Figure 15.

![Fig. 13. Reference path for different values of \( u_1 = \{-0.2, -0.1, 0, 0.1, 0.2\} \), with \( u_2 = 0 \).](image)

C. Reducing the Effect of Noise

It is important to characterize the noise affecting the measured cost and constraints. The estimation of experimental
conversion to cost and constraint values
\text{MA optimization algorithm}
conversion to reference path

Fig. 15. RTO block diagram.

gradients can be very badly affected by high-frequency noise, which in turn can lead to unreasonable optimized inputs. The average line tension and the minimum altitude per path cycle are shown in Figure 16, while the kite followed the same reference path for 6 minutes. The minimum height is relatively consistent, with variations of mostly \( \pm 1 \) m, illustrating the consistency of the path-following controller. However, the average line tension is extremely variable, ranging from 60 kg to 120 kg, that is, with up to 35 % noise. This would be considered extreme noise in the RTO literature. This noise is not induced by variable controller performance, as the kite follows almost exactly the same path during each cycle. Rather, the noise is caused by wind variations. The wind speed measured at the ground station during the same experiment is shown in Figure 17. The measured wind speed varies significantly over time, and there is a rough correlation between the variations in the measured wind speed and the line tension. High-frequency variations in the wind direction also affect the line tension. The path is centered with respect to the

average wind direction (the x-axis in Figure 3 is continually realigned with the average measured wind direction over the past 15 minutes). However, the measured wind direction can easily veer by up to 15 degrees for shorter periods. In theory, the measured wind speed and direction could probably be used to scale the average measured line tension in order to reduce the ‘noise’. However, in this work, a simpler approach was adopted.

The simple solution to dealing with noise is to partially remove it via averaging. The effect of averaging the line tension over \( N_{\text{avg}} = 7 \) path cycles is apparent in Figure 18. This reduces the noise to a manageable (if still quite high) level. The expense of this is that the RTO algorithm only iterates every 7 path cycles, and thus proceeds more slowly.

Fig. 17. The wind speed measured at the ground station during the experiment shown in Figure 16 (blue), and the average wind speed per reference-path cycle (black dots).

Fig. 16. The kite’s altitude and the measured line tension during 6 minutes following a constant reference path (blue). The minimum attained altitude and the average line tension per path cycle (black).

Fig. 18. No averaging (dashed with crosses), averaging over \( N_{\text{avg}} = 7 \) path cycles (solid with circles). During this experiment, the kite followed a constant reference path.
D. RTO Results

The RTO algorithm’s performance was tested over several days of experiments. Depending on the conditions, the attained optimal reference path was quite different. For example, in light winds, the path tended to be much wider (large values of $u_2$) than in stronger winds. This makes practical sense as the kite becomes far more sluggish and unresponsive to steering at low wind speeds; hence, only gently curving paths can be followed efficiently. However, in order to verify that the RTO algorithm performs correctly, it is useful to test it many times under near-identical conditions, namely, with constant wind speed and constant controller parameters. Unfortunately, meeting the first condition is complicated by the ever-changing wind conditions. Nonetheless, at some point, several hours of continuous experiments were carried out in relatively constant wind conditions without any adjustments being made to the experimental setup, and these results are presented next.

Figure 19 shows the line tension and the kite’s altitude during 30 minutes of autonomous flight. During the first 7 minutes (13 iterations) and the last 5 minutes (10 iterations), the kite follows a constant reference path that is rather high and narrow, resulting in a low average line tension, about 80 kg. The RTO algorithm markedly improved the average line tension, increasing it to about 135 kg, during the intermediate 22 minutes that are depicted with a grey-shaded area on Figure 19. The decision variables during this experiment are shown in Figure 20. The zig-zagging behavior is in part caused by the MA algorithm exciting the process in order to estimate experimental gradients (exploration), and in part due to the effect of wind-induced noise. Luckily, the fact that the RTO algorithm is constrained to taking small steps means that, while it zig-zags, on average it moves towards the plant optimum. And so, the limited step-size introduces a certain robustness to noise.

Next, in order to verify that the RTO algorithm did indeed converge to a neighborhood of the plant optimum, an experimental study was carried out to see how the average line tension varies with respect to the path followed by the kite. This consisted of measuring the average line tension over 10 minutes for a variety of different paths. The resulting surface is shown in Figure 21. Note that the average measured wind speed was relatively constant during the entire experimental study. Also, it was carried out immediately after the experiment shown in Figure 19, and the conditions were essentially the same. It can be observed that the maximum attainable average line tension is about 130-140 kg. It is interesting to note that the nominal optimal solution that was calculated using the model, corresponding to $u_1 = u_2 = 0$, results in an average line tension of about 115 kg. Hence, relying only on model-based optimization would have resulted in an optimality loss of about 15-20% in this case. Figure 22 superimposes the MA algorithm’s path on this contour. It is readily seen that, despite the zig-zagging behavior, the algorithm converges to the neighborhood of the plant optimum.

VI. DISCUSSION AND CONCLUSIONS

The two-layer control and optimization algorithm for kites described in this paper was designed for, and implemented
on, a particular experimental setup. However, we believe that a number of useful conclusions can be drawn regarding the control and optimization of kite-power systems in general.

Firstly, while additional measurements are certainly useful, good path-following control (of a kite flying roughly crosswind) can be achieved using only measurements of the kite’s position. However, delay must be either eliminated or compensated for, as due to the kite’s speed, even a small delay can cause oscillations around the reference path.

Secondly, the average line tension varies significantly depending on the path followed by the kite, to the extent that serious attention should be given to the kite’s path if efficiency is to be maximized. It is questionable whether model-based optimization alone can calculate an efficient path. Indeed, in this study, using the model’s optimal path would have resulted in a 15-20% lower average line tension compared to the plant optimal path, despite calibrating the model using experimental data. It is true that a relatively simplistic model was used, yet more complex models, although they would certainly lead to a smaller loss in average line tension, may not be practical for two main reasons: (i) they are often too computationally demanding for path optimization, and (ii) they cannot easily account for changes in wind conditions. The solution presented here is to use RTO to constantly update the reference path in order to continuously react to wind and system variations.

How do the algorithms described here apply to the main commercial situations? Currently, there exist many different systems, and the methods described in this paper are more suited to some than to others.

- **Modeling**: In terms of dynamic modeling, the approach in this paper is suitable for flexible power kites. It does not assume any particular kite geometry or size, and it is equally applicable to a 3-m$^2$ or 300-m$^2$ kite. As it is a very intuitive and simple model, it can easily be modified to take into account effects such as nonlinear steering dynamics, a variable angle-of-attack or a varying tether length. This modeling approach is indeed well suited for flexible kites, as more accurate detailed models of flexible kites have yet to reach maturity. Rigid kites, on the other hand, can benefit from the much more advanced modeling techniques available from classical flight dynamics.

- **Control**: The path-following controller is, in theory, applicable to any power kite flying in crosswind mode. The low-level velocity-angle controller must be designed and tuned on a system-to-system basis, as different kites tend to have different steering behavior. Depending on the steering dynamics and the system delay, this may require an integral term or delay compensation. The geometric guidance algorithm, on the other hand, could be applied to almost any kind of power-kite system.

- **Real-time optimization**: The RTO algorithm is readily applicable to ship-towing and fixed-length tether crosswind electricity generation, but would need to be significantly extended to handle pumping-cycle electricity generation. The MA framework itself is very general and has also been applied to a variety of chemical processes. However, the particular formulation of the optimization problem (the choice of the cost and constraint functions and of the decision variables) must be adapted to each specific problem. Maximization of thrust (the component of line tension in a desired direction) for ship-towing is very similar to the maximization of line tension dealt with in this article. Only a minor change would need to be made to the definition of the cost function and the same decision variables (width and height of the reference path) could be used. The center point of the reference path would be determined by the direction of travel, and when this makes a significant angle with the apparent wind, it might be useful to introduce a further decision variable to represent the asymmetry of the figure-of-eight path. In fixed-length-tether crosswind kite power with turbines on the wing, instantaneous electricity production is almost proportional to tether tension, and so the optimization algorithm from this paper could be applied with little modification. The reference path should be centered directly downwind, either using wind measurements or using the approach described in [28].

Pumping-cycle electricity generation is unfortunately a
little trickier to deal with, as due to the variation in the tether length, each figure-of-eight will be subject to wind at different altitudes. The cost function could be the average electricity produced over an entire pumping cycle, and the decision variables could be the height and width relative to the tether length and either the tether tension or the tether reel-out speed.

Many additions and improvements to this work could be envisaged. In terms of control, if increased path-following accuracy is required, there is no doubt that a well-designed feed-forward signal could improve the velocity control loop. As the reference path is repetitive, another promising option would be to use run-to-run control, such as iterative learning control [44]. The guidance algorithm could be generalized to motion on a sphere using geodesic distances and angles, following the approach in [15]. At the expense of some mathematical abstraction, this framework is advantageous as it does not approximate the quarter-sphere upon which the kite flies using a plane, which results in a certain distortion.

Perhaps the most promising avenue is pitch control, using the kite’s second degree of freedom. Up until now, controller development has focused on the “steering” input, which is usually the difference between the lengths of the rear lines. However, the kite’s pitch angle relative to the lines can also be adjusted, usually by changing the length of the front line relative to that of the rear lines. This additional degree of freedom is typically used for manual kite control; it directly influences the kite’s angle of attack, which has a strong effect on the line tension, the kite’s speed, and the kite’s turning behavior. The first step in exploiting this degree of freedom is to establish experimentally validated models describing the influence of the kite’s angle of attack, which are currently not available.

In terms of real-time optimization, there is much room for improvement by using wind measurements. If the influence of wind speed and direction on the line tension can be computed using a model, and subtracted from the observed variations, then the line-tension variations due to path changes alone can be more easily detected.

ACKNOWLEDGMENT

The authors would like to thank Nikitas Rontsis, Ioannis Lymperopoulos, Jose Afonso, Yannick Poffet, Damien Benoit, Paul Bertusi and Enea Martinoli for their contributions to this project. We would also like to thank Corey Houle, Rolf Luschinger and the entire Swiss Kite Power group for many years of open and inspiring collaboration.

Sean Costello finished a PhD at EPFL in Lausanne, Switzerland in 2015. His research areas are control, real-time optimization and kite power. He is currently working on sensor fusion and machine control with Leica Geosystems.

Grégory François received a PhD degree in Process Systems Engineering from EPFL Lausanne, Switzerland in 2004 and an Habilitation Thesis also in Process Systems Engineering from the University of Lorraine, France, in 2014. His research interests include modeling, control, optimal control and optimization of static and dynamical systems, with application to a wide range of systems including chemical, polymerization, energy and fuel-cell systems.

Dominique Bonvin is Director of the Automatic Control Laboratory of EPFL in Lausanne, Switzerland. He received his Diploma in Chemical Engineering from ETH Zurich and his Ph.D. degree from the University of California, Santa Barbara. He served as Dean of Bachelor and Master studies at EPFL between 2004 and 2011. His research interests include modeling, identification and optimization of dynamical systems.