Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Automatic trajectory tracking control of kites

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfillment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

JORN H. BAAYEN

Delft, the Netherlands
March 2011

MSc THESIS APPLIED MATHEMATICS

"Automatic trajectory tracking control of kites"

JORN H. BAAYEN

Delft University of Technology

Daily supervisor	**Responsible professor**
Dr. D. Jeltsema	Prof. J.H. van Schuppen

Other thesis committee members

Dr. R. Schmehl

Dr. Q. P. Chu

Prof. W. J. Ockels

March 2011

Delft, the Netherlands

Acknowledgements

First of all, Mom and Dad, for always being there.

My supervisors at DIAM, prof. Jan van Schuppen and Dr. Dimitri Jeltsema, for help with formulating my results.

My advisor at ASSET, Dr. Roland Schmehl, for the rigor.

Dr. Ping Chu, for advice and encouragement.

Dr. Jacob van der Woude and Dr. Rik Lopuhaä, for help in making my unusual study track possible.

Stefan de Groot, Aart de Wachter, Barend Lubbers, and Rolf van der Vlugt, for the insight.

My housemate Tjerk Vreeken, for listening, encouragement and advice.

All the other people of the 10th floor: thank you for all discussions and ideas, for turning 2010 into such a fruitful year.

Contents

Chapter 1

Introduction

In 1980 Loyd wrote a seminal paper exploring the possibility of generating electrical power using the pulling force of tethered airfoils, i.e., kites [30]. During the oil glut of the 1980s, however, interest in wind energy dropped. It was only during the turn of the 21$^{\text{st}}$ century that interest in kite power (and traction) picked up again with the work of Meijaard, Ockels, Schwab and Diehl [32, 33, 8].

On December 17th, 2007, the MS *Beluga Skysails* departed from Bremerhaven, Germany, to Guanta, Venezuela. The 132 m long container ship – see Figure 1.1 – was equipped with the first commercial application of kite traction: a 160 m^2 computer-controlled kite used to reduce fuel consumption by 15-20%. Since then several more ships have been equipped with kites, and, at the time of writing, more installations are planned.

Figure 1.1: MS *Beluga Skysails* with its 160 m^2 kite (`http://skysails.info`).

Six years earlier, in 2001, the first academic investigation into automatic control of kites had appeared [8]. In this work, Diehl had investigated the possibility of using non-linear model predictive control (MPC) for trajectory tracking control of kites, initiating a line of research that continues to this day [8, 10, 24, 21, 39, 12, 22]. Unfortunately, non-linear, and in this case even non-convex, MPC suffers from several drawbacks. The strict constraints on the time available for solving the on-line optimal controls problem inhibit convergence guarantees, and the lack of a convex problem structure places uncertainties on the properties of any optima that the optimizer converges to. As a consequence, the stability of MPC kite controllers remains unproven.

In this work, we propose an alternative trajectory tracking control paradigm that is purely analytical. Kites generally have a single control input available for steering, and in the following we reduce the dimensionality of the trajectory tracking problem to obtain a single-input single-output control problem. Subsequently, we derive a non-linear control law and prove that it achieves tracking.

Before describing our control algorithm, however, we introduce the concept of kite power in more detail. This is followed by a review of various kite modeling paradigms in the second chapter. Derivation of the control law follows in chapter three. In the fourth and final chapter of this thesis we validate our assumptions with simulations against various kite models.

Chapter 2

Kite power

A kite is a tethered airfoil. As an airfoil moves through the air, the pressure difference between the underside and the upper side of the wing generates a force. The component of this force that is perpendicular to the airflow is called the lift force, and assuming a positive angle between the airfoil and the airflow, this force points upwards. Any lift force exceeding the gravity force of the kite and the tension force of the tether can be used for work.

2.1 The principle of crosswind power

The lift force produced by an airfoil at specific flow conditions (including the angle of attack[1]) is given by

$$F_L = \frac{1}{2}\rho V_a^2 A C_L,$$

where ρ is the density of air, V_a is the airspeed[2], A is the surface area of the wing, and C_L is the unitless lift coefficient for the current flow conditions [1].

Figure 2.1: Crosswind figure eight trajectory flown using the *Laddermill* prototype kite power plant. The wind vector points along the road towards the horizon.

In 1980, Loyd observed that the airspeed, and hence the pulling force of the kite, can be increased by flying the kite in directions perpendicular to the wind vector [30]. To maximize the pulling force of a kite, then, one flies so-called *crosswind* trajectories. Examples of such

---

[1] The angle of attack is the angle between a reference line on the airfoil and the airflow.

[2] The airspeed V_a is the velocity of the air past the wing. It can be computed as $V_a = \|\mathbf{v}_a\|$, with $\mathbf{v}_a = \mathbf{v}_w - \mathbf{v}$, and where $\mathbf{v}_a$ is the *apparent* or *aerodynamic* velocity vector, $\mathbf{v}_w$ is the wind vector, and $\mathbf{v}$ in the velocity vector of the wing.

trajectories include the crosswind figure eight and the crosswind circle; see Figures 2.1 and 2.2. The figure eight is commonly preferred because it avoids torsion in the tether: after the tether is wound one turn around itself, it is then unwound in the other direction.

2.2 Wind energy at high altitudes

The power density of the wind is given by

$$P_w = \frac{1}{2}\rho V_w^3,$$

with ρ the density of air and V_w the wind speed [30, 6]. In other words, the power that can be extracted from the wind scales with the cube of the wind speed.

Now winds at high altitude are stronger than they are at low altitude [2]. Based on 27 years of data Archer and Caldeira [2] find that the highest wind power densities lie at an altitute of $10,000$ m above Japan and eastern China, the eastern coast of the United States, southern Australia, and north-eastern Africa, with median values greater than $10,000$ W/m^2. This value contrasts starkly with a typical value at low altitute in the United Kingdom: 140 W/m^2 [31].

The optimal height of $10,000$ m would be hard to reach using the steel tower of a conventional wind turbine. A kite, on the other hand, might rise to this altitude. The power of the wind might then be transferred down to the ground, either mechanically, or electrically.

2.3 Energy conversion concepts

Several concepts have been developed to exploit crosswind and high-altitude kite power. We review the concepts of airborne turbines, pumping motion, as well as the concept of direct traction.

2.3.1 Airborne turbines

One way to transform wind energy into electrical energy is by mounting turbines on the kites. The generated power is then transmitted down to earth through an electrically conducting tether. As of 2011, three Californian startups, *Joby Energy Inc.*, *Magenn Power Inc.*, and *Makani Power Inc.*, are exploring this concept.

Figure 2.2: The *Makani* kite flying crosswind circles (`http://makanipower.com`).

2.3.2 Pumping motion

The *Laddermill* prototype kite power plant developed at the Institute for Applied Sustainable Science, Engineering and Technology (ASSET), Delft University of Technology, uses a kite to pull a tether from a drum driving a generator to produce power. The operational mode consists of two distinct phases: a reeling-out phase pulling the tether from the drum producing power and a reeling-in phase which consumes energy; see Figure 2.3. It is desired to control the kite in such a manner that a net power production remains between the reeling-out and reeling-in phase.

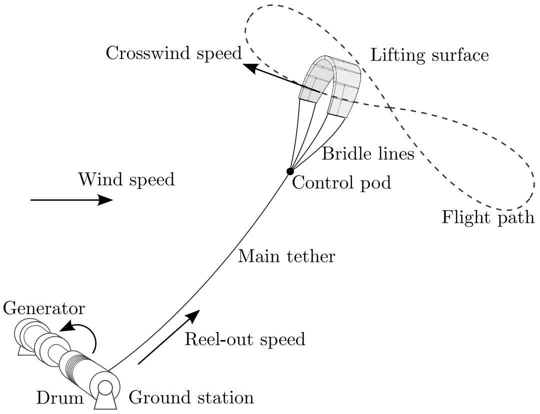

Figure 2.3: The Laddermill during the reel-out phase [7].

Since 2007, the Italian company *Kite Gen Research SRL* has been investigating the same concept. As of 2011, Kite Gen Research are in the process of constructing its first stationary kite power plant.

2.3.3 Traction

The simplest, and as of 2011 only commercially succesful, application of kite power is direct traction. The firm *SkySails GmbH & Co. KG* in Hamburg, Germany, sells kite power systems for ships. The pulling force of a SkySails kite acts on the bow of the ship, thereby pulling it forward and upward out of the water; see Figure 1.1. The company website (`http://skysails.info`) claims that the SkySails system provides fuel savings of up to 50%.

Chapter 3

Modeling

In the present chapter we investigate several approaches to the modeling of kites and tethers.

3.1 Kite modeling

In this section, we review several approaches to kite modeling, in order of increasing complexity. We start out by reviewing the steering mechanism of interest, after which we proceed to an extension of a commonly used point-mass model, briefly touch upon research into rigid-body modeling, and finally discuss multi-body models of flexible kites.

3.1.1 Steering mechanism

The control mechanism developed at the ASSET institute consists of a control pod suspended under the kite. The control pod is connected to the drum on the ground using a single tether and four bridle lines lead from the control pod to the kite. The pod contains two servo motors, one controlling the length difference of the two back lines, the *steering lines*, and the other motor controlling the overall length of the steering lines. By changing the length difference of the steering lines the kite deforms; see Figure 3.1. The aerodynamics in turn generate steering yaw moments. A detailed explanation is available in [5]. Regulation of the total length of the steering lines provides some control over the attack angle of the kite itself. This is known as the *powering* and the *depowering* of the kite.

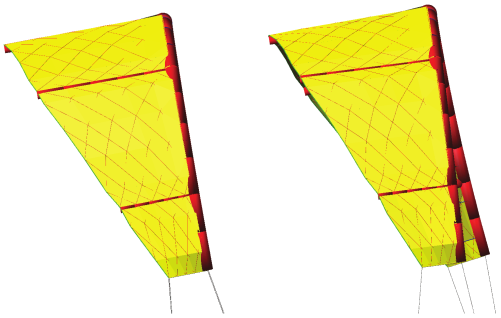

Figure 3.1: Kite before and after applying a steering input [5].

In the remainder of this thesis we will restrict our attention to flexible arc-shaped kites with a control mechanism as described.

3.1.2 Point-mass models

We present an extension of the point-mass kite model discussed in [8, 20, 21, 24, 12]. Our extension is threefold: we modify the definition of the aerodynamic reference frame to account for sagging of the tether, we change the steering mechanism to be based on sidewards steering forces instead of on the rolling of the lift vector (i.e., banking), and we use a lift curve instead of a single lift constant.

Let $\mathbf{r}$ denote the vector connecting the ground tether attachment point to the kite. Let $\mathbf{p}$ denote the unit vector pointing down from the kite towards the control pod; see Figure 3.2. We refer to this vector $\mathbf{p}$ as the *pod vector*.

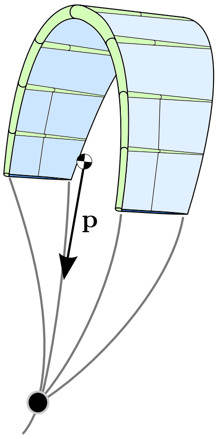

Figure 3.2: The unit vector $\mathbf{p}$ pointing down towards the control pod.

We express the aerodynamic forces acting on the kite in an aerodynamic reference frame. Let $\mathbf{v}_a = \dot{\mathbf{r}} - \mathbf{v}_w$ denote the apparent wind vector, where $\mathbf{v}_w$ represents the environmental wind. Let $\mathbf{e}_x$ be the unit vector collinear with the apparent wind vector $\mathbf{v}_a$. We assume that the kite cannot roll with respect to the pod vector, and so we must have $\mathbf{e}_y$ such that it is parallel to $\mathbf{p} \times \mathbf{e}_x$. Choose $\mathbf{e}_y$ to be collinear with $\mathbf{p} \times \mathbf{e}_x$ and let $\mathbf{e}_z := \mathbf{e}_x \times \mathbf{e}_y$; Figure 3.3 illustrates the construction.

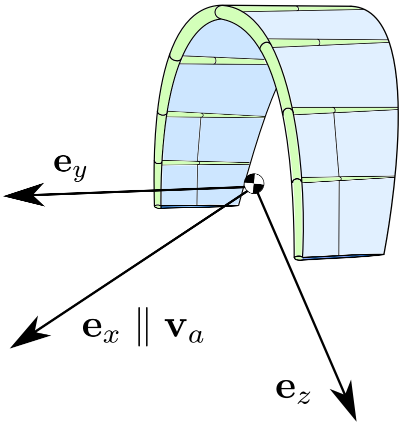

Figure 3.3: Aerodynamic reference frame.

In expressing the magnitude of the forces, we will make use of unitless lift coefficients. Let $C_L : \alpha \mapsto C_L(\alpha)$ be the unitless lift curve with $\alpha := \operatorname{acos}(\mathbf{p} \cdot \mathbf{e}_z) + \theta_p$, where θ_p is the power setting, and let $C_S : u \mapsto C_S(u)$ be the unitless steering force curve. We model the drag force as the sum of a zero-lift drag coefficient and a parabolic approximation of induced drag [1]:

$$C_D = C_{D_0} + kC_L^2.$$

The constant k is defined by $k^{-1} = \pi Ae$, with aspect ratio $A = b^2/S$, span efficiency factor e, projected surface area S, and wingspan b. It follows that the aerodynamic forces are given by

$$\begin{aligned} F_x &= -\frac{1}{2}\rho\|\mathbf{v}_a\|^2 SC_D(\alpha), \\ F_y &= -\frac{1}{2}\rho\|\mathbf{v}_a\|^2 SC_S(u), \\ F_z &= -\frac{1}{2}\rho\|\mathbf{v}_a\|^2 SC_L(\alpha), \end{aligned}$$

where ρ is the density of air.

In our model, the steering control input u causes the kite to accelerate in the direction of the Y-axis, which in turn causes an increase in the relevant component of the apparent wind vector $\mathbf{v}_a$. The aerodynamic reference frame rotates along with the apparent wind vector, and in this way the effects of the steering yaw moments are emulated. The model is completed by considering gravity, and by connecting it to a tether model.

In our simulations we choose a projected surface area of $S = 11$ m^2, a wing span of $b = 6$ m, an Oswald factor of $e = 0.7$ and a mass of $m = 13$ kg. The zero-lift drag coefficient is given by $C_{D_0} = 0.075$. The lift curve C_L, modeled on data from [34] for a NACA 0018 airfoil in a flow with Reynolds number $\mathrm{Re} = 2 \times 10^6$ using B-splines, is given in Figure (3.4). Due to the current lack of more specific data, we also set $C_S = C_L$. This decision is based on the observation that one can view the tips of the kite as vertically positioned airfoils, with a local attack angle controlled using the steering control input u.

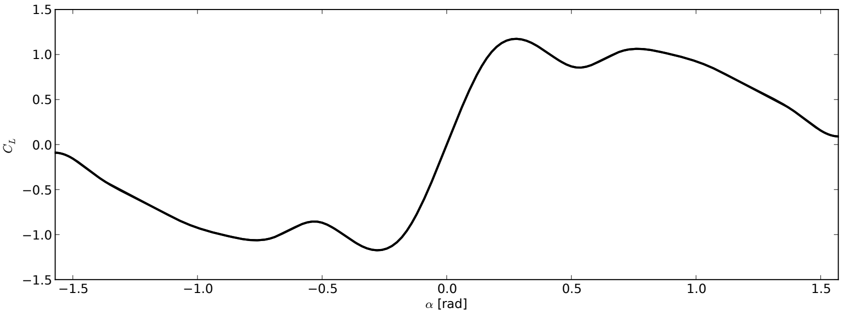

Figure 3.4: Lift curve modeled on data from [34] for a NACA 0018 airfoil at $\mathrm{Re} = 2 \times 10^6$.

We stress that this is a simple, heuristic model that is unlikely to capture more than the basic aspects of kite flight. The model is meant as a test bed for controller development where besides kite-like behaviour, a low computational cost of simulations is a primary requirement. The multi-body models described below are fully validated, but computationally expensive to simulate.

3.1.3 Rigid-body models

In [7] system identification techniques are used to reduce a multi-body model of a flexible kite to a rigid-body approximation. This process is frayed with complications, and as a result we will restrict our attention to the simple point-mass model described earlier, as well as to the full multi-body models described in the next section.

3.1.4 Multi-body models

In [5] a methodology for developing multi-body models of flexible kites is developed. Such models consists of three kinds of building blocks: cables, inflatable tubes, and foils.

A cable is modeled as a chain of discrete elements connected by joints. The motions of the cable elements are damped using a damping coefficient that depends on the tension in the cable. Aerodynamic drag forces act on the centers of the cable elements, and elasticity can optionally be modeled by the inclusion of spring-dampers in between the cable elements.

An inflatable tube is modeled as a string of rigid elements connected by spherical joints. The bending and torsion behaviour of the beam is determined by torque vectors imposed on the joints.

The foil elements model the fabric of the kite and consist of leading edge tubes to which wire elements are attached in the chordwise direction; see Figure 3.5. The wire elements consist of discrete elements connected by joints. The chordwise wires are connected diagonally by springs; also see Figure 3.5. Aerodynamic forces and moments based on computational fluid dynamics calculations act on the joints in the wire elements.

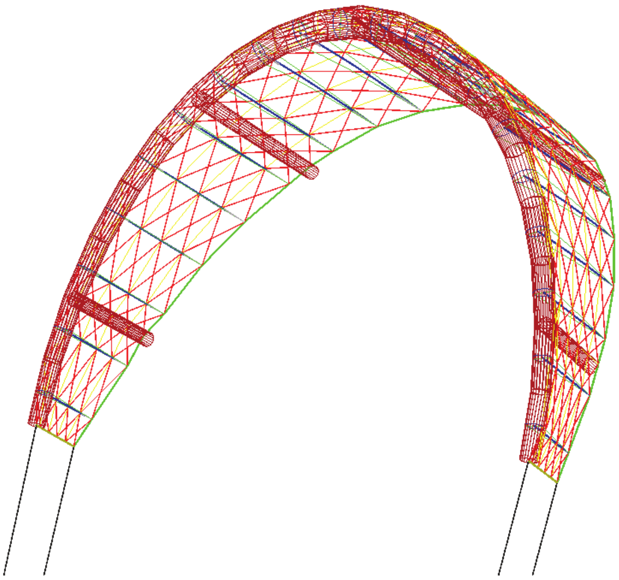

Figure 3.5: A multi-body model of an arc-shaped kite [5].

While these models are computationally expensive to simulate (it takes about an hour to simulate 2 minutes of flight on an Intel Core Duo computer), simulation is still a lot cheaper than finite element simulations [5]. At the same time, the multi-body models accurately model the physical mechanism of asymmetric deformation that underpins the steering behaviour of arc-shaped kites [5].

3.2 Tether modeling

In this section, we review several approaches to tether modeling, in order of increasing complexity. We start out by touching upon constraint and spring-damper based models of tethers that do not take sagging effects into account, after which present a model of a flexible tether consisting of point-masses connected by spring-dampers. Finally we briefly mention the existence of flexible tether models consisting of strings of rigid bodies.

3.2.1 Constraint models

The point-mass kite model described in [8, 24, 21, 12] uses a spherical coordinate system. In this way the tether is modeled implicitly as constraining the kite to a sphere. While computationally cheap to simulate, effects of tether sag are neglected.

3.2.2 Spring-damper models

In [7], the tether is modeled as a single spring-damper. Similarly to the constraint models the single spring-damper neglects tether sagging effects, but it is computationally cheap to simulate.

3.2.3 Point-mass models

We obtain a simple model of a flexible tether by connecting a string of point-masses and spring-dampers. To model cable drag we include aerodynamic drag forces. The magnitude of the aerodynamic drag forces that act on the springs is proportional to the lengths of the springs, and this force is distributed over the neighbouring point-masses. All point-masses have gravity acting on them. The cable is connected to the kite by considering the kite as one of the cable point-masses; see Figure 3.6.

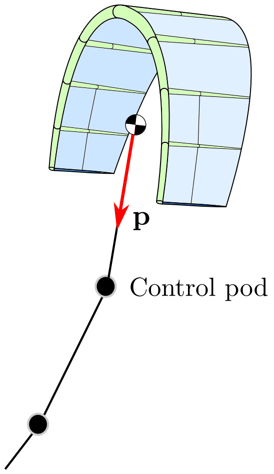

Figure 3.6: The control pod as a point-mass in the tether.

To make this precise, we fix a spring and describe the forces acting on its neighbouring point-mass nodes. First of all we note that if $l < l_0$, that is, if the tether element is slack, we set the aerodynamic as well as the spring-damper forces to zero. If $l \geq l_0$, then the forces assume the form described below.

Aerodynamic forces. Fix a spring and let $\mathbf{d}$ be a unit vector pointing from the top incident mass to the bottom incident mass or the other way around. Let $\mathbf{v}_a$ be the average aerodynamic velocity of the incident masses, and let

$$\mathbf{v}_{a,\perp} := \mathbf{v}_a - (\mathbf{v}_a \cdot \mathbf{d})\mathbf{d},$$

the component of $\mathbf{v}_a$ that is perpendicular to $\mathbf{d}$. We model the drag force as acting in the opposite direction of $\mathbf{v}_{a,\perp}$ with magnitude

$$F_a = \frac{1}{2}\rho\|\mathbf{v}_{a,\perp}\|^2 A C_D^t$$

with cable element surface area A and a unitless drag coefficient C_D^t.

Spring forces. Fix either incident node and let $\mathbf{d}$ point towards the mass in question. The spring force acts along the vector $\mathbf{d}$ and is given by Hooke's law

$$F_s = -k(l - l_0)$$

where l is the distance between the incident masses, l_0 is the rest length of the spring, and k is the spring constant. Let k_σ be the spring constant of the entire tether and let k_0 be the *unit spring constant*. That is, $k_\sigma := k_0/l_\sigma$, where l_σ is the length of the entire tether. Connecting springs in series we have

$$\frac{1}{k_\sigma} = \sum_{i=1}^{n} \frac{1}{k_i},$$

where $k_i = k_0/l_{0,i}$ are the spring constants of the cable elements. It follows that $k = k_0/l_0$.

Damping forces. The damping force acts on the velocity component $v_\parallel := \mathbf{v} \cdot \mathbf{d}$ in the direction of the vector $\mathbf{d}$. Its magnitude is given by

$$F_d = -cv_\parallel,$$

where c is the damping constant. Similarly to the above we have that, given the *unit damping constant* c_0, $c = c_0/l_0$.

Varying tether length. To increase the length of the tether, the rest length of the first spring, spring A, is increased. If the new length exceeds a set threshold, then it is split in two and a new point-mass is added in between. The new node is then given a length and velocity in such a way that leaves the forces at the upper end of spring A unchanged.

This force balance constraint can be achieved in the following way. Let $C = l_0'/l_0$, where l_0' is the new rest length of spring A. Let

$$l' = C(l - l_0) + l_0'$$

be the new rest length of spring A, and let

$$v' = Cv.$$

be the velocity of the new node. It is easy to verify that force balance is obtained if the new node is set to be at a distance l' from the top node along the original spring.

Similarly, to decrease the length of the tether, the rest length of the first spring is decreased. Once it is below a certain threshold it is removed and its rest length is added to the next spring. The node that previously was in between the two springs is deleted.

For our simulations we use the characteristics of a *Dyneema* cable with a 2 mm radius (`http://matbase.com`). Dyneema has a density of about 960 kg/m^3 and its Young's modulus is about 55 GPa, whence the unit spring coefficient is

$$k_0 = 55 \times 10^9 \cdot (4 \times 10^{-3})^2 = 880000 \text{ N}.$$

The unit damping constant $c_0 = 30000$ Ns was chosen such as to give stable behaviour in simulation. The value of the drag coefficient, $C_D^t = 1.22$, is taken from [26] for a smooth cylinder at an attack angle of $\pi/2$.

To simulate the presence of a control pod, the next point-mass down the cable from the kite is given an extra mass of 7 kg; see Figure 3.6. The pod vector **p** is the unit vector pointing down from the kite in the direction of the control pod node.

We stress that this is a simplistic model only meant to capture the basic sagging behaviour observed in kite tethers. The purpose of the model is to verify that our controller operates reliably with flexible, sagging tethers.

3.2.4 Rigid-body models

A validated multi-body approach to tether modeling is explored in [5], where the cable is modeled as described in Section 3.1.4.

Chapter 4

Control

The design of automatic controllers for kite power systems has been a subject of ongoing research; see for instance [8, 10, 24, 38, 14, 15, 16, 39, 20, 21, 12, 7]. In this chapter, we briefly review existing concepts, and introduce our contribution: a novel, non-linear control law with stability guarantees. The section on our contribution is based on a paper submitted to the *IET Journal of Control Theory & Applications*.

4.1 Previous research

In this section, we briefly review previous research on trajectory tracking control of kites.

4.1.1 Linear control

An obvious approach to trajectory tracking control of kites is to apply linear control methods to the linearized kite dynamics. Use of a linear quadratic regulator (LQR) has been investigated in [10, 38, 39]. In [10], however, it is shown that such a controller cannot handle large disturbances (as is to be expected, since a linearization tends to be valid in a small region around the linearization point only). A kite controller should be able to keep a kite in the air in face of large wind gusts, and for this reason research into LQR control of kites has seemingly been abandoned in favour of non-linear control methods.

4.1.2 Model predictive control

A model predictive controller uses a model of the controlled system to solve a discretized optimal control problem at every time step. The first of the computed controls is implemented, and on the next time step the procedure is restarted: a new optimal control problem is solved, and the new first computed control is implemented. In this way the changing state of the system is taken into account, i.e., the loop is closed.

Starting with [8] a decade of research has been invested in non-linear model predictive control (MPC) of kites [8, 10, 24, 21, 39, 12, 22]. Diehl shows that the resulting controller can handle larger disturbances than a linear quadratic regulator [10]. In [10] it is also shown how the discretized optimal control problems arising in MPC can be solved in real-time. In [12], on the other hand, the optimal control problems problems are solved off-line. The on-line controller then interpolates between stored control values to determine the appropriate closed-loop response.

Unfortunately, the MPC method as applied to trajectory tracking control of kites has several drawbacks. To our knowledge, no convex formulation of the kite control problem is known. The non-convexity implies that the optimizer may converge to different local optima, depending on the initial estimates. This leads to uncertainty about controller performance. Closely related is the current lack of a stability proof for a tracking MPC kite controller[1].

A second drawback is that a succesful implementation of a non-interpolating non-linear MPC controller requires use of advanced, real-time capable optimization methods. Without expert knowledge of the inner workings of these optimization methods, debugging an MPC controller can be hard. Storing pre-computed controls, on the other hand, requires a lot of memory. Suppose that the control law depends on the position of the kite (3 coordinates), its velocity (3 coordinates), and the airspeed (1 coordinate). If we partition each dimension into a hundred intervals, then we would need to store $100^7 = 10^{14}$ control inputs. At a single byte per control input, this would add up to 90 terabytes of data. In practice one may be able to discard irrelevant parts of state space, but a formidable memory requirement remains.

A third complication is the need for a fully identified and reasonably accurate kite model. As shown in [7] identification of such a model is difficult for flexible kites. Without a good, real-time optimization capable model, however, an MPC controller computes commands that are inappropriate for the actual system.

The discussed three drawbacks have prompted the development of our contribution, an easily implemented and provably stabilizing non-linear control law.

4.1.3 Neural network control

In [14, 15, 16] an adaptive neural network is developed and trained for kite control tasks. While a trained network can be understood by analyzing its connection structure, the behaviour of the controller remains relatively opaque in comparison to alternative control approaches – a neural network controller is like a "black box". This opacity of operation, as well as the lack of stability results, compel us to invest more confidence in alternative methods. Nevertheless a neural network controller may "devise" novel approaches to kite control, and in this way one may gain novel insights by observing a neural network controller in action.

4.2 Non-linear adaptive tracking control

In this section we introduce our contribution: a new, non-linear adaptive control law with stability guarantees.

First of all, we introduce a trajectory tracking concept based on the angle between the velocity vector and the horizontal. This concept is restricted to the motion of the kite in directions perpendicular to the vector connecting the earth tether attachment point and the kite, leaving control of the reeling in and reeling out of the tether up to a separate controller. To get a first intuitive picture of this idea, let us for a moment project the crosswind figure eight trajectory of a kite onto the plane perpendicular to the wind vector. As a kite traces out a figure eight on this plane, the angle the velocity vector makes with the horizontal oscillates between certain minimum and maximum values; see Figures 4.1 and 4.2.

Instead of measuring this angle directly we use the more general notion of *turning angle* [17] from elementary differential geometry (see the Appendix). This angle is not restricted to $[-\pi, \pi]$ and it is continuous. It is cumulative in the sense that multiple turns around the origin of the velocity vector will result in a multiple of 2π, but winding backwards will reduce the turning angle by the same amount for every turn. This property makes the turning angle well suited for control purposes. By considering the arc length as independent variable, the turning angle

---

[1] In [9] nominal stability is proved for a real-time non-linear model predictive control scheme, under the assumption that the discretized optimal control problems have unique optimal solutions.

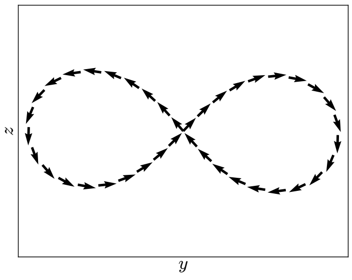

Figure 4.1: Tangent vectors along path.

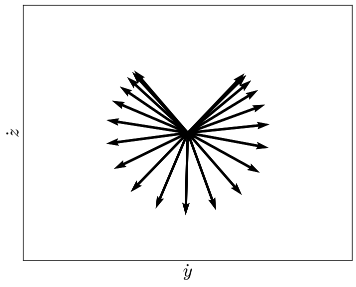

Figure 4.2: Tangent vectors translated to the origin.

– together with initial conditions – becomes a representation of the image of a curve that is independent of parametrization.

Secondly, we derive an adaptive turning angle tracking controller that does not need a full aerodynamic kite model. Instead it only uses the control derivatives relating control input increments to steering force increments; the rest of the required data is measured directly. We provide a stability proof for the resulting controller and demonstrate its performance with simulations.

4.2.1 Geometric modeling and system

We are interested in controlling flight in directions perpendicular to that of the radial vector $\mathbf{r}$ connecting the earth tether attachment point and the kite. The control of the tether and of the power lines is left to a separate controller. In order to derive a control system that is invariant of the tether length, we project the flight trajectory onto the upper half of the unit sphere centered on the earth tether attachment point. This yields the projected path $\boldsymbol{\gamma} := \mathbf{r}/\|\mathbf{r}\|$.

Geometric model

As mentioned in the introduction, we will control the turning angle of the kite trajectory. In order to be able to compute the turning angle on the upper half of the unit sphere we need a coordinate system. For this note that the coordinate patch

$$\mathbf{p} : [0, 2\pi) \times [0, \pi/2] \to \mathbb{R}^3, \quad (v, w) \mapsto (\cos w \cos v, \cos w \sin v, \sin w)\,.$$

yields a continuous one-to-one (except for the pole) correspondence between the upper half of the sphere and the set $[0, 2\pi) \times [0, \pi/2]$. The vector fields $\partial\mathbf{p}/\partial v$ and $\partial\mathbf{p}/\partial w$ are orthogonal, and by normalizing them we obtain an orthonormal basis of the tangent plane,

$$\mathbf{e}_1 := (-\sin v,\, \cos v,\, 0), \quad \mathbf{e}_2 := (-\cos v \sin w,\, -\sin v \sin w,\, \cos w), \tag{4.1}$$

as illustrated in Figure 4.3. We choose our orientation to be compatible with $\mathbf{e}_1 \times \mathbf{e}_2$.

The vector field $\mathbf{e}_1$ has no vertical component, which intuitively makes it a suitable reference for the definition of a turning angle. To make this precise, we apply Liouville's theorem [17] for which we need the curvatures of the coordinate system. The coordinate curves $w \mapsto \mathbf{p}(v, w)$ are parts of great circles, whence their geodesic curvature $(\kappa_g)_2$ is zero. The curves $v \mapsto \mathbf{p}(v, w)$, however, have geodesic curvature $(\kappa_g)_1 = \sin w$. By Liouville's theorem, the turning angle of the projected kite trajectory with respect to the vector field $\mathbf{e}_1$ is given by the integral equation

$$\theta[\boldsymbol{\gamma}] := \theta[\boldsymbol{\gamma}, \mathbf{e}_1] = \int_{\boldsymbol{\gamma}} \kappa_g[\boldsymbol{\gamma}] - (\kappa_g)_1 \cos\theta[\boldsymbol{\gamma}, \mathbf{e}_1]\, ds + \theta[\boldsymbol{\gamma}, \mathbf{e}_1]_0, \tag{4.2}$$

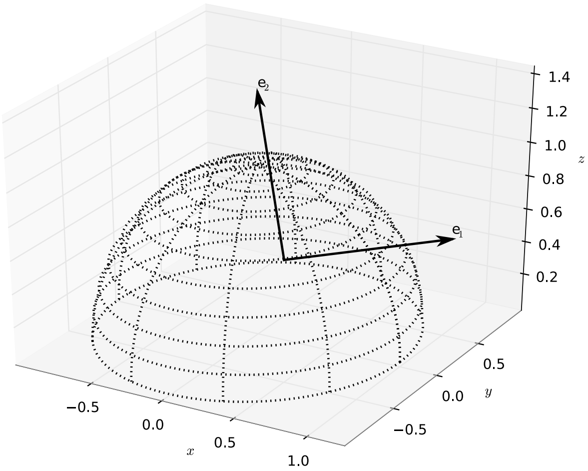

Figure 4.3: Coordinate system on the upper half of the unit sphere.

where $\kappa_g[\boldsymbol{\gamma}]$ is the geodesic curvature of the projected kite trajectory and $\theta[\boldsymbol{\gamma}, \mathbf{e}_1]_0$ is the initial value of the turning angle. The geodesic curvature of the projected kite trajectory can be calculated as

$$\kappa_g[\boldsymbol{\gamma}] = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\|\dot{\boldsymbol{\gamma}}\|^3}, \tag{4.3}$$

with $\dot{x}$ and $\dot{y}$ the coordinates of the velocity of the projected trajectory, expressed in the $(\mathbf{e}_1, \mathbf{e}_2)$-basis of the tangent plane.

Path length

The projected velocity of the kite may sometimes have a component perpendicular to the target trajectory $\boldsymbol{\gamma}_t$. If we zig-zag around the target trajectory and straighten out the traversed path we obtain a longer arc length than if we would straighten out the target trajectory. We need to take this difference into account, for otherwise we would move through the target turning angle quicker than intended. For this reason we introduce a corrected path length that disregards the perpendicular component.

Let $\mathbf{T}_t := \cos\theta[\boldsymbol{\gamma}_t]\mathbf{e}_1 + \sin\theta[\boldsymbol{\gamma}_t]\mathbf{e}_2$, where $(\mathbf{e}_1, \mathbf{e}_2)$ is the orthonormal basis of the tangent plane at $\boldsymbol{\gamma}_t$ given by (4.1). This unit vector field is tangent to the target trajectory, and taking the inner product $\dot{\boldsymbol{\gamma}} \cdot \mathbf{T}_t$ we obtain the component of the projected velocity tangential to the target trajectory. This motivates the following definition.

Definition 4.2.1. *We refer to the integral*

$$s_c := \int \dot{\boldsymbol{\gamma}} \cdot \mathbf{T}_t \, dt$$

as the corrected path length.

Control system

Our objective is to track the given target trajectory $\boldsymbol{\gamma}_t$ which, as hinted as previously, we will do by controlling the turning angle $\theta[\boldsymbol{\gamma}]$ to track the turning angle of the target trajectory. We will show that we can achieve this objective using a single steering control input. This way we construct a one-dimensional single-input, single-output tracking problem.

In order to arrive at a set of equations that are affine in their control input we base our discussion on steering input *increments* $\tilde{u} := u - u^*$. Central to this linearizing approach is the following assumption, which holds when controlling the kite at sufficiently high frequency.

Assumption 4.2.1. *The acceleration $\ddot{\boldsymbol{\gamma}}$ is locally affine in feasible*[2] *control increments $\tilde{u}$. That is,*

$$\left|\frac{\partial^3 \gamma}{\partial u \partial t^2}\right| \gg \left|\frac{\partial^4 \gamma}{\partial u^2 \partial t^2}\tilde{u}\right|.$$

Linearizing the derivative of the turning angle in the control input u we find, using (4.2), (4.3) and the definition of the corrected path length, that

$$\begin{aligned}\frac{d\theta[\gamma]}{dt} &= \left.\frac{d\theta[\gamma]}{dt}\right|_{u^*} + \left.\frac{\partial^2\theta[\gamma]}{\partial u \partial t}\right|_{u^*}\tilde{u} \\ &= \left.\frac{d\theta[\gamma]}{ds}\frac{ds}{dt}\right|_{u^*} + \left.\frac{\partial^2\theta[\gamma]}{\partial u \partial s}\frac{ds}{dt}\right|_{u^*}\tilde{u} \\ &= \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\|\dot{\boldsymbol{\gamma}}\|^2} - \boldsymbol{\gamma}\cdot\mathbf{e}_z \cos\theta[\gamma]\|\dot{\boldsymbol{\gamma}}\| + \frac{\dot{x}\frac{\partial \ddot{y}}{\partial u} - \frac{\partial \ddot{x}}{\partial u}\dot{y}}{\|\dot{\boldsymbol{\gamma}}\|^2}\tilde{u},\end{aligned} \tag{4.4}$$

where $\boldsymbol{\gamma}\cdot\mathbf{e}_z$ denotes the (earth) vertical component of the projected trajectory. In deriving our control loop in the next section we will make use of the direct coupling between the control increment $\tilde{u}$ and the derivative of the turning angle. This principle is similar to the incremental nonlinear dynamic inversion approach for control of aircraft as developed by Sieberling et al. [35].

For the derivatives $\partial\ddot{x}/\partial u$ and $\partial\ddot{y}/\partial u$, representing the sensitivities of the accelerations to changes in the steering input, we need a model. Let $r = \|\mathbf{r}\|$ and note that

$$\ddot{\boldsymbol{\gamma}} = \frac{d^2}{dt^2}\left(\frac{\mathbf{r}}{r}\right) = \frac{\ddot{\mathbf{r}}}{r} - \frac{\ddot{r}\mathbf{r}}{r^2} - 2\frac{\dot{r}\dot{\mathbf{r}}}{r^2} + 2\frac{\dot{r}^2\mathbf{r}}{r^3}.$$

Expressed in the tangential frame the first two coordinates of the radial vector $\mathbf{r}$ are zero. As a result, and due to the fact that only the control derivatives of the accelerations are non-zero, we see that the sensitivities, expressed in the tangential frame, are given by

$$\begin{aligned}\frac{\partial \ddot{x}}{\partial u} &= \frac{\rho v_a^2 S C_x^t}{2mr} \\ \frac{\partial \ddot{y}}{\partial u} &= \frac{\rho v_a^2 S C_y^t}{2mr}.\end{aligned} \tag{4.5}$$

Here C_x^t and C_y^t are the unitless *control derivatives* expressed in the tangential frame, ρ is the density of air, v_a is the airspeed, S is the projected surface area of the airfoil, and m is the combined mass of the kite and the control pod. Let $\mathbf{C} = [C_x \;\; C_y \;\; C_z]$ denote the control derivatives expressed in a body-fixed reference frame.

Kites deform during flight and at any point in time the deformation depends on the form the kite had previously. Hence the deformation of the kite is *path-dependent.* In order to model the control derivatives, we assume path-dependent deformation to have a – for control purposes – negligible influence on the steering aerodynamics.

The weight of the control pod suspended under the kite inhibits the rolling motion of the airfoil with respect to the vector connecting the mass centers of the kite and the pod; see Figure

---

[2] We call a control input increment feasible when the actuator can implement it within a single sampling interval.

3.2. This implies that the tether force acts in a direction perpendicular to the sidewards steering direction, whence the sidewards steering forces are unaffected by the cable dynamics. This, together with our discussion on path-dependent deformation, motivates the following assumption.

Assumption 4.2.2. *The unitless control derivatives* $\mathbf{C}$*, expressed in the body-fixed reference frame, are a function of the current steering input* u^**.*

Note that this assumption can be relaxed by including the dependence on additional state variables. Dependencies that may need to be added include tether length, if the control derivative in the direction of the tether is found to vary significantly with line length, as well as the power setting of the kite.

We approximate the function $u^* \mapsto \mathbf{C}$, mapping states to control derivatives, using B-spline networks as in [37]. Let $\mathbf{w}_i$ denote the weights and $\mathbf{b}$ the basis functions of the B-spline network for the ith control derivative with $i \in \{x, y, z\}$, so that the ith B-spline network is given by $C_i(u^*) = \mathbf{w}_i \cdot \mathbf{b}(u^*)$. The number of basis functions used for the spline networks must be chosen in such a way as to avoid problems with under- or overparametrization. This can be done using a trial and error procedure.

By continuously measuring the linear velocity, linear acceleration, airspeed and attitude of the kite, we obtain a nonlinear system of the form

$$\frac{d\theta[\boldsymbol{\gamma}]}{dt} = f(\theta[\boldsymbol{\gamma}], t) + B(u^*, t)\tilde{u}. \tag{4.6}$$

The measurements are implicit in the time-dependence of the functions f and B, which are given by

$$f = \left.\frac{d\theta[\boldsymbol{\gamma}]}{dt}\right|_{u^*} = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\|\dot{\boldsymbol{\gamma}}\|^2} - \boldsymbol{\gamma} \cdot \mathbf{e}_z \cos\theta[\boldsymbol{\gamma}]\|\dot{\boldsymbol{\gamma}}\|$$

and

$$B = \left.\frac{\partial^2\theta[\boldsymbol{\gamma}]}{\partial u \partial t}\right|_{u^*} = \frac{\dot{x}\frac{\partial \ddot{y}}{\partial u} - \frac{\partial \ddot{x}}{\partial u}\dot{y}}{\|\dot{\boldsymbol{\gamma}}\|^2}. \tag{4.7}$$

By choosing our system in this way the need for a full aerodynamic kite model vanishes.

From (4.4) and (4.5) we see that the function B is linear in the control derivatives $C_i(u^*) = \mathbf{w}_i \cdot \mathbf{b}(u^*)$, that is,

$$B = \sum_i \lambda_i \mathbf{w}_i \cdot \mathbf{b} \tag{4.8}$$

for some coefficients $\lambda_i = \lambda_i(t) \in \mathbb{R}$. This shows that the system (4.6) is affine in both the control input increment $\tilde{u}$ as well as in the control derivative spline weights $\mathbf{w}_i$, a property that will allow us to design an adaptive control loop with relative ease in the next section.

4.2.2 Control

In this section we design our control loop and control derivative estimators by means of a control-Lyapunov approach. Our approach is closely related to those described in [37, 3, 27, 28, 23].

Control problem

Our objective is to track a given reference projected kite trajectory, and in the following we investigate ways to formalize this problem.

To this end, a first idea is to use the difference between the actual and the target turning angle $\theta[\boldsymbol{\gamma}] - \theta[\boldsymbol{\gamma}_t]$ as our notion of tracking error. The value of the turning angle at any point in time, however, does not encode where in space the kite is. It may then happen that turning-angle steering errors accumulate and that our trajectory starts to drift. In order to remedy this situation we include another term encoding the signed distance from the target path, which we will call the *signed linear tracking error*:

$$\mu(\boldsymbol{\gamma}, \boldsymbol{\gamma}_t) := \|\boldsymbol{\gamma} - \boldsymbol{\gamma}_t\| \operatorname{sgn}\left(J\dot{\boldsymbol{\gamma}}_t \cdot (\boldsymbol{\gamma} - \boldsymbol{\gamma}_t)\right),$$

where $J(x, y) = (-y, x)$ denotes a $\pi/2$ counterclockwise rotation in the tangent plane at the point $\boldsymbol{\gamma}_t$. This number has a magnitude equal to the linear tracking error $\|\boldsymbol{\gamma} - \boldsymbol{\gamma}_t\|$, and the sign indicates whether the vector $\boldsymbol{\gamma} - \boldsymbol{\gamma}_t$ points to the left or to the right of the target trajectory. This encoding makes it clear in which direction the heading of the kite needs to be corrected.

We let our cost function consist of a weighted sum of the error in turning angle and the signed linear tracking error:

$$e := P(\theta[\gamma] - \theta[\gamma_t]) + I\mu(\boldsymbol{\gamma}, \boldsymbol{\gamma}_t), \tag{4.9}$$

with weights $P, I \in \mathbb{R}^+$. The choice of I as symbol for the μ-gain is intentional: conceptually μ can be seen as the integral of the component of the projected velocity that is perpendicular to the target trajectory. In order to track the reference trajectory we must find a control law that renders $e \to 0$.

The steering dynamics of a kite may subtly change over time due to weather patterns and wear. In order to automatically adapt to these changes we aim for an adaptive control approach that maintains and updates an internal representation of the steering control derivatives. We will denote this internal representation as $\hat{\mathbf{w}}_i$ with $i \in \{x, y, z\}$.

Summarizing, we formulate the following control problem:

Problem 4.2.1. *Find a steering control law and a control derivative B-spline weight estimator update law that such that $e \to 0$ as $t \to \infty$.*

Control design

In this section we propose steering control and estimator update laws to solve Problem 4.2.1 using a control-Lyapunov approach.

Proposition 4.2.1. *Consider the system (4.6). The control law*

$$\tilde{u} := -\frac{1}{P\sum_i \lambda_i \hat{\mathbf{w}}_i \cdot \mathbf{b}}\left[P\left(\left.\frac{d\theta[\gamma]}{dt}\right|_{u^*} - \frac{d\theta[\gamma_t]}{dt}\right) + I\frac{d\mu}{dt}(\boldsymbol{\gamma}, \boldsymbol{\gamma}_t) + Ke\right], \tag{4.10}$$

with gain $K \in \mathbb{R}^+$, and the control derivative B-spline weight estimator update law

$$\frac{d\hat{\mathbf{w}}_i}{dt} := \Gamma e P \lambda_i \tilde{u} \mathbf{b}, \tag{4.11}$$

with adaptation gain $\Gamma \in \mathbb{R}^+$, render the Lyapunov function

$$V := \frac{1}{2}e^2 + \frac{1}{2}\Gamma^{-1}\sum_i \|\hat{\mathbf{w}}_i - \mathbf{w}_i\|^2 \tag{4.12}$$

decreasing along the flow.

Proof. The time-derivative of the Lyapunov function V is given by

$$\frac{dV}{dt} = e\frac{de}{dt} + \Gamma^{-1}\sum_i (\hat{\mathbf{w}}_i - \mathbf{w}_i)\cdot\frac{d\hat{\mathbf{w}}_i}{dt} \tag{4.13}$$

with, using (4.4),

$$\frac{de}{dt} = P\left(\left.\frac{d\theta[\gamma]}{dt}\right|_{u^*} + \left.\frac{\partial^2\theta[\gamma]}{\partial u\partial t}\right|_{u^*}\tilde{u} - \frac{d\theta[\gamma_t]}{dt}\right) + I\frac{d\mu}{dt}(\gamma,\gamma_t).$$

The estimator update laws (4.11) cancel out all terms with $\mathbf{w}_i$. Indeed, replacing these into (4.13), we obtain, using (4.7) and (4.8), that

$$\frac{dV}{dt} = e\left[P\left(\left.\frac{d\theta[\gamma]}{dt}\right|_{u^*} - \frac{d\theta[\gamma_t]}{dt}\right) + I\frac{d\mu}{dt}(\gamma,\gamma_t)\right] + \sum_i eP\lambda_i\tilde{u}\hat{\mathbf{w}}_i\cdot\mathbf{b}. \tag{4.14}$$

Finally, substituting the control law (4.10) into (4.14), we see that

$$\frac{dV}{dt} = -Ke^2.$$

The gain $K \in \mathbb{R}^+$ renders V strictly decreasing whenever $e \neq 0$. □

We obtain a different perspective on the control law (4.10) if we assume that $\hat{\mathbf{w}}_i = \mathbf{w}_i$. Setting $I = K = 0$ and replacing the control law into the system (4.6), we obtain that

$$\frac{d\theta[\gamma]}{dt} = \frac{d\theta[\gamma_t]}{dt}.$$

This indicates that (4.10) is a generalization of a feedback linearizing control law.

In the previous analysis we neglected rate and magnitude constraints of the control signal. Let $\tilde{u}_0$ denote the control increment as computed using (4.10) and let $\tilde{u}$ denote the control increment implemented by the actuator. With the actuated control signal the Lyapunov function (4.12) is no longer guaranteed to be decreasing and the parameter estimation may diverge. In order to remedy this situation we introduce the *modified tracking error* as in [13],

$$e_m := e - \zeta,$$

with

$$\frac{d\zeta}{dt} := -K\zeta + P\left.\frac{\partial^2\theta[\gamma]}{\partial u\partial t}\right|_{u^*,\hat{\mathbf{w}}_i}(\tilde{u} - \tilde{u}_0), \tag{4.15}$$

providing a correction for the influence of the actuator limitations, where the second derivative is evaluated using the estimated weights $\hat{\mathbf{w}}_i$. Note that in the absence of actuator constraints $e_m \to e$.

In order to render the modified tracking error to zero, we adjust the estimator update law (4.11) to be proportional to e_m. The associated modified Lyapunov function is – by construction – decreasing along the flow, independently of whether or not $\tilde{u} = \tilde{u}_0$:

Proposition 4.2.2. *Consider the system (4.6). Any control law and the modified control derivative B-spline weight estimator update law*

$$\frac{d\hat{\mathbf{w}}_i}{dt} := \Gamma e_m P\lambda_i\tilde{u}\mathbf{b} \tag{4.16}$$

render the modified Lyapunov function

$$V_m := \frac{1}{2}e_m^2 + \frac{1}{2}\Gamma^{-1}\sum_i \|\hat{\mathbf{w}}_i - \mathbf{w}_i\|^2 \tag{4.17}$$

decreasing along the flow.

Proof. The time-derivative of the modified Lyapunov function (4.17) is given by

$$\frac{dV_m}{dt} = e_m \frac{de_m}{dt} + \Gamma^{-1} \sum_i (\hat{\mathbf{w}}_i - \mathbf{w}_i) \cdot \frac{d\hat{\mathbf{w}}_i}{dt} \tag{4.18}$$

with, using (4.15) and (4.4),

$$\begin{aligned} \frac{de_m}{dt} &= K(e - e_m) - P \left. \frac{\partial^2 \theta[\boldsymbol{\gamma}]}{\partial u \partial t} \right|_{u^*, \hat{\mathbf{w}}_i} (\tilde{u} - \tilde{u}_0) \\ &+ P \left(\left. \frac{d\theta[\boldsymbol{\gamma}]}{dt} \right|_{u^*} + \left. \frac{\partial^2 \theta[\boldsymbol{\gamma}]}{\partial u \partial t} \right|_{u^*} \tilde{u} - \frac{d\theta[\boldsymbol{\gamma}_t]}{dt} \right) + I \frac{d\mu}{dt}(\boldsymbol{\gamma}, \boldsymbol{\gamma}_t). \end{aligned}$$

The modified estimator update laws (4.16) cancel out all terms with $\mathbf{w}_i$. Indeed, replacing these into (4.18) we obtain, using (4.7) and (4.8), that

$$\begin{aligned} \frac{dV_m}{dt} &= e_m \left[K(e - e_m) - P \left. \frac{\partial^2 \theta[\boldsymbol{\gamma}]}{\partial u \partial t} \right|_{u^*, \hat{\mathbf{w}}_i} (\tilde{u} - \tilde{u}_0) \right] \\ &+ e_m \left[P \left(\left. \frac{d\theta[\boldsymbol{\gamma}]}{dt} \right|_{u^*} - \frac{d\theta[\boldsymbol{\gamma}_t]}{dt} \right) + I \frac{d\mu}{dt}(\boldsymbol{\gamma}, \boldsymbol{\gamma}_t) \right] \\ &+ \sum_i e_m P \lambda_i \tilde{u} \hat{\mathbf{w}}_i \cdot \mathbf{b}. \end{aligned} \tag{4.19}$$

Noting that $\tilde{u} = (\tilde{u} - \tilde{u}_0) + \tilde{u}_0$ and substituting the canonical control law (4.10) as $\tilde{u}_0$ into (4.19) we see that

$$\frac{dV_m}{dt} = -K e_m^2.$$

The gain $K \in \mathbb{R}^+$ renders V_m strictly decreasing whenever $e_m \neq 0$. □

Control performance

In this section we investigate the performance of the proposed control and estimator update laws. In the case that $\tilde{u} = \tilde{u}_0$, the function V defined in (4.12) is decreasing along the flow. This implies convergence of the kite trajectory $\boldsymbol{\gamma}$ to the target trajectory $\boldsymbol{\gamma}_t$.

Theorem 4.2.1. *Consider the system (4.6), the control law (4.10) and the estimator update laws (4.16). Assume that there is a* $t_0 > 0$ *such that for all* $t > t_0$

$$\|\dot{\boldsymbol{\gamma}}(t)\| > 0$$

holds, so that for $t > t_0$ *the kite trajectory is a regular curve. Additionally assume that for all* $t > t_0$ *the geometric condition*

$$\left. \frac{d\ddot{\boldsymbol{\gamma}}}{du} \right|_t \nparallel \mathbf{r}(t),$$

i.e., the condition that the vectors $d\ddot{\boldsymbol{\gamma}}/du|_t$ *and* $\mathbf{r}(t)$ *are not parallel, is satisfied, so that the control law is well-defined. Then for any initial condition and for any reference trajectory the kite trajectory converges to the target:*

$$\lim_{t \to \infty} \boldsymbol{\gamma}(t) = \boldsymbol{\gamma}_t(t).$$

Proof. LaSalle's Invariance Principle [29, 27] applied to the Lyapunov function V shows that $e \to 0$. □

We proceed to show that, even when $\tilde{u} \neq \tilde{u}_0$, the modified parameter estimation process is stable in the sense that the weight estimation errors remain bounded:

Theorem 4.2.2. *Consider the system (4.6) and the estimator update laws (4.16). Assume that there is a* $t_0 > 0$ *such that for all* $t > t_0$ *the conditions*

$$\|\dot{\gamma}(t)\| > 0$$

and

$$\left.\frac{d\ddot{\gamma}}{du}\right|_t \not\parallel \mathbf{r}(t)$$

are satisfied. Then for any initial estimates, any reference trajectory and any control increments $\tilde{u}$*, the estimator errors*

$$\|\hat{\mathbf{w}}_i(t) - \mathbf{w}_i(t)\|, \quad i \in \{x, y, z\}$$

are bounded, and the bound decreases in time for $t > t_0$.

Proof. Boundedness follows from the definition of the Lyapunov function V_m:

$$\|\hat{\mathbf{w}}_i(t) - \mathbf{w}_i(t)\| \leq \sqrt{2\Gamma V_m(t)}.$$

Proposition 4.2.2 shows that the bound decreases in time. □

We do not, however, have a guarantee that the estimates will converge to the true values of the weights. The modified Lyapunov function V_m becomes stationary once $e_m = 0$ and at this point the estimation error will not be able to decrease further.

Chapter 5

Simulation results

In the present chapter we investigate the performance of our controller with two kite models. In Section 5.1 we describe the implementation of the controller, and in Section 5.2 section we define our reference trajectory. In Section 5.3 section we investigate controller performance with the point-mass model from Chapter 2. In Section 5.4 we demonstrate performance with a multi-body model of a flexible kite, thereby validating Assumption 4.2.2 on the functional dependence of the control derivatives. In Section 5.5 we investigate the adaptivity properties of our controller, and finally, in Section 5.6, we compare the performance of our controller with the performance of a linear control law.

5.1 Implementation

We implemented our controller in the C++ programming language. For linear algebra we use the *Eigen* template library [18], and for integration we link to the CVODE integrator from the *Sundials* library [19].

For large values of the estimator update gain Γ the estimator update laws (4.16) can become stiff differential equations. For this reason we use the backward differentiation formula (BDF) [11] implicit integration mode of the CVODE integrator (with Newton iteration). For optimal results we run the integrator in variable step size mode. Because in our framework kite measurements become available at a certain frequency (e.g., 10 Hz), we have to interpolate between available measurements for the variable step size integration. We find that linear interpolation performs well for this purpose.

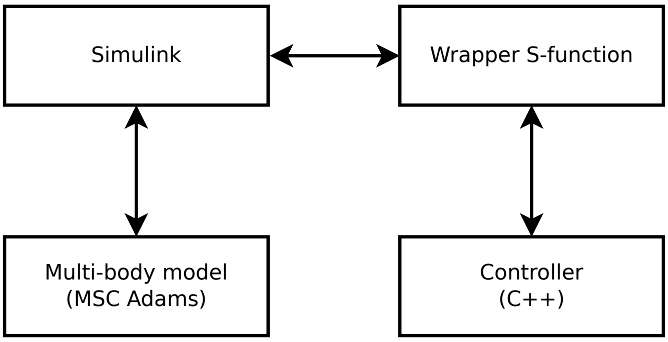

Figure 5.1: The coupling between the multi-body model and the controller.

While the controller is coupled directly to a C++ implementation of the point-mass model, the coupling with the multi-body model is more elaborate. The multi-body model is implemented in the *MSC Adams* package. MSC Adams, run in server mode, can be controlled from the *Simulink* software. In order to bring our controller into the picture we developed a Simulink S-function wrapping our controller, as well as a Simulink model connecting the controller and the multi-body model. This configuration is illustrated in Figure 5.1.

5.2 Reference trajectory

In order for the tracking objective to be independent of the velocity we treat the corrected path length – as opposed to time – as the independent variable. Our reference turning angle is

$$\theta[\boldsymbol{\gamma}_t](s_c) := A\left[\cos\left(\frac{2\pi}{L}s_c\right) - 1\right] + \theta_0, \tag{5.1}$$

where $A \approx 2.40483$ is a root of the 0th Bessel function of the first kind, $J_0(A) = 0$, and $L = 4/3$ is the length of the projected target trajectory. This turning angle generates a figure eight trajectory. For the planar notion of turning angle, the constant A being a root of the appropriate Bessel function is a necessary and sufficient condition for the turning angle (5.1) to result in an L-periodic trajectory. To see this, note that in the planar case the target tangent vector at s_c is $\mathbf{T}_t(s_c) = (\cos\theta[\boldsymbol{\gamma}_t](s_c), \sin\theta[\boldsymbol{\gamma}_t](s_c))$, and that

$$\begin{aligned}
x(L) &= x(0) + \int_0^L \cos\theta[\boldsymbol{\gamma}_t](s_c)\,ds_c \\
&= x(0) + \int_0^L \cos\left(A\cos\left(\frac{2\pi}{L}s_c\right)\right)\cos(\theta_0 - A) - \sin\left(A\cos\left(\frac{2\pi}{L}s_c\right)\right)\sin(\theta_0 - A)\,ds_c \\
&= x(0) + L\cos(\theta_0 - A)J_0(A), \\
y(L) &= y(0) + \int_0^L \sin\theta[\boldsymbol{\gamma}_t](s_c)\,ds_c \\
&= y(0) + \int_0^L \sin\left(A\cos\left(\frac{2\pi}{L}s_c\right)\right)\cos(\theta_0 - A) + \cos\left(A\cos\left(\frac{2\pi}{L}s_c\right)\right)\sin(\theta_0 - A)\,ds_c \\
&= y(0) + L\sin(\theta_0 - A)J_0(A).
\end{aligned}$$

The third constant in the reference turning angle (5.1), $\theta_0 = 0.834029$, is the initial condition, which also corrects for the curvature of the sphere. It was found using a root-finding procedure. The resulting trajectory is an idealization of trajectories flown using the Laddermill prototype kite power plant. Figure 2.1 shows a trajectory flown using the Laddermill, and Figure 4.1 plots our idealization.

5.3 Point-mass model

In this section we report on the performance of our controller with the point-mass model of a kite described in Section 3.1.2, tethered using a 300 m 25 spring point-mass and spring-damper cable model as described in Section 3.2.3.

The wind vector points in the direction of the earth X-axis with magnitude 6 m/s. The controller gains are tuned – using a trial and error procedure – to $K = 3.0$, $P = 0.1$, $I = 0.1$ and $\Gamma = 10^5$, the steering rate is limited to 0.05 rad/s and the magnitude of the steering input to 0.015 rad. The power setting is fixed at $\theta_p = 0.01$ rad. Every control derivative estimator B-spline is of the 3rd order with 21 uniformly spaced knots (interpolation points), and the controller runs at 10 Hz.

Simulation results for a run on constant line length without turbulence are plotted in Figures 5.2 and 5.4. Part of a trajectory with a tether reel out velocity equal to a third of the wind speed is shown in Figure 5.3.

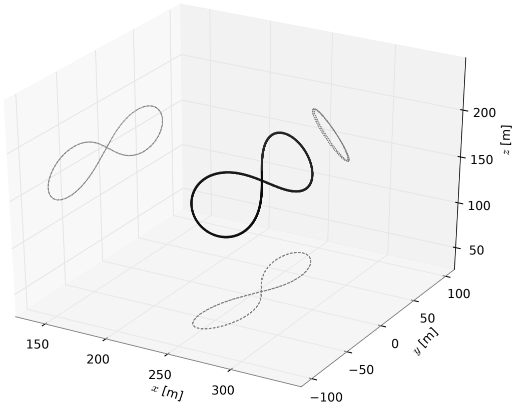

Figure 5.2: Three overlapping figure eight trajectories.

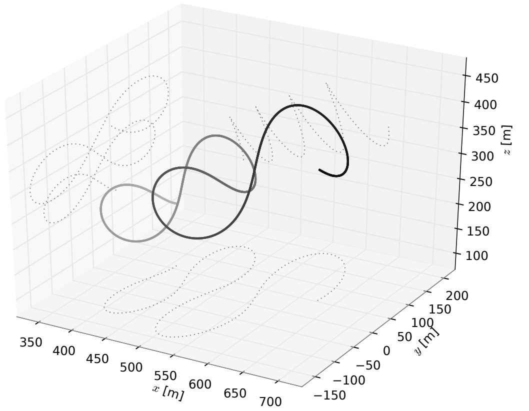

Figure 5.3: Part of a trajectory during tether reel out.

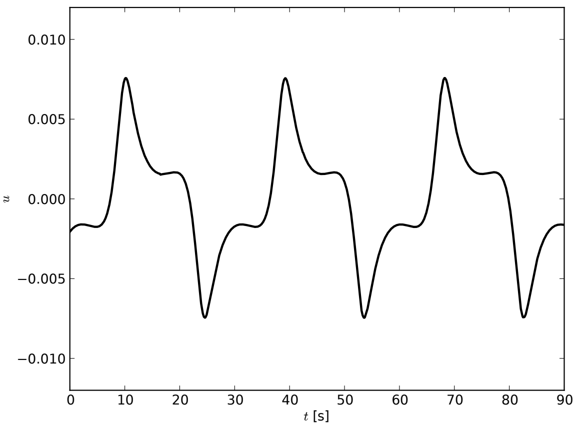

Figure 5.4: Control input for the trajectory shown in Figure 5.2.

In order to quantify the controller performance under the influence of turbulence we consider the mean linear tracking error

$$J := \frac{1}{T}\int_0^T \|\gamma - \gamma_t\|\, dt. \tag{5.2}$$

Figure 5.5 plots the mean linear tracking error J for a single cycle as a function of simplified Dryden [4] turbulence intensity for precise control derivative estimates. The correlation rate δ determines the spectral density of the wind.

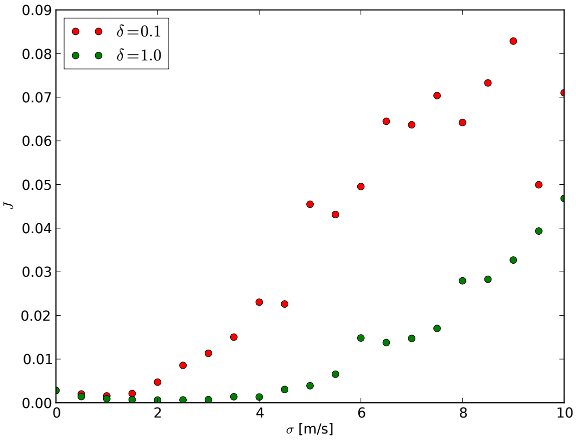

Figure 5.5: Mean linear tracking error as function of turbulence intensity σ and correlation rate δ, where $\sigma_{\text{lo}} = \sigma_{\text{la}} = \sigma_{\text{ve}} = \sigma$. Every data point is an average of 100 simulations.

For design of wind turbines a turbulence intensity of 20 % is commonly assumed [6]. Using this rule of thumb on our 6 m/s wind speed, we obtain $\sigma = 0.20 \cdot 6 = 1.2$ m/s. At this level of turbulence, the tracking error remains bounded by a value of 0.005. Scaling this number with the distance between earth tether attachment point and the kite we obtain a bound of 1.5 m, which is a quarter of the wingspan of the simulated kite.

5.4 Multi-body model

In this section we investigate the performance of our control law with a multi-body model of a 15 m^2 flexible arc-shaped kite developed using the tools from [5]. In order to treat the multi-body kite model as a rigid body, as called for by the framework of our controller, we use the position, velocity and acceleration of its center of gravity. De Groot computes the inertia tensor and angular momentum relative to the center of gravity and uses this to define an attitude [7].

The kite model is controlled by changing the length difference of the steering lines. Initial estimates of the control derivatives are obtained by fitting the linear model

$$C_i = C_{i,0} + C_{i,u}u, \quad i \in \{x, y, z\},$$

to the aerodynamic forces observed in manually controlled simulations using the method of least squares.

In our simulation, the wind vector points in the direction of the earth X axis with magnitude 6 m/s, the controller gains are tuned – using a trial and error procedure – to $K = 3.0$, $P = 0.1$, $I = 0.03$ and $\Gamma = 10^3$, the steering rate is limited to 0.4 m/s and the magnitude of the steering input to 1 m. Every control derivative estimator B-spline is of the 3rd order with 21 uniformly spaced knots, and the controller runs at 10 Hz.

Simulation results for a run on a line of 300 m without turbulence are shown in Figures 5.6 and 5.7. The observed tracking validates Assumption 4.2.2 on the functional dependence of the control derivatives.

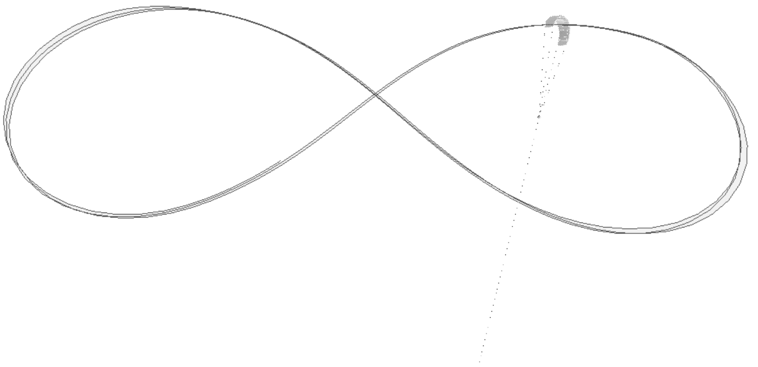

Figure 5.6: Controlled flight of the multi-body model.

Detailed analysis of performance under turbulence is not currently available due to the fact that the multibody simulations become an order of magnitude slower when introducing turbulence into the wind model, to the extent where a full turbulence analysis would take about a year to run on our Intel Core Duo system. Nevertheless results from single simulations look promising; see Figure 5.8 for the steering input from a simulation with simplified Dryden turbulence at intensities $\sigma_{\mathrm{lo}} = \sigma_{\mathrm{la}} = \sigma_{\mathrm{ve}} = 1.2$ m/s and correlation rate $\delta = 1.0$.

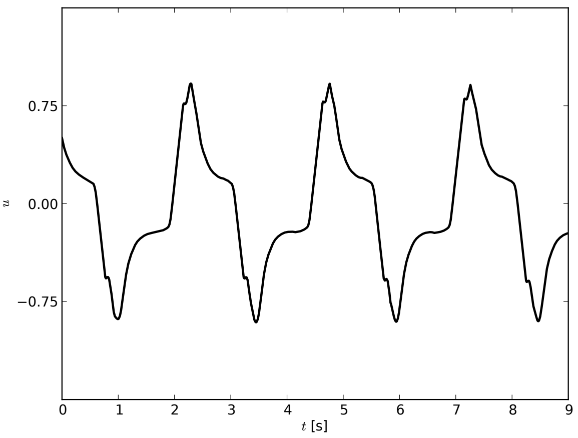

Figure 5.7: Control input for the trajectory shown in Figure 5.6.

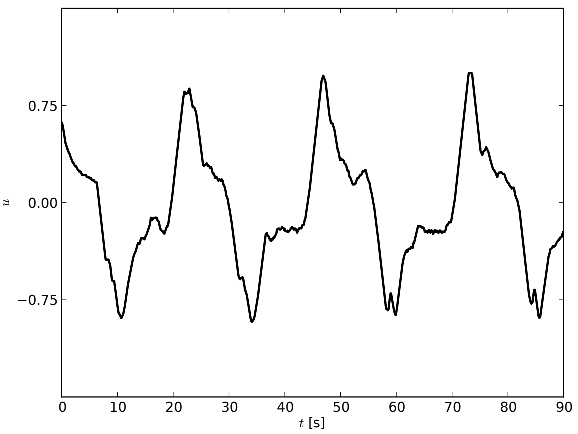

Figure 5.8: Control input for turbulent simulation ($\sigma = 1.2$ m/s, $\delta = 1.0$).

5.5 Adaptivity

While the theory is unambiguous, the numerical convergence of parameter estimates in an adaptive controller can be problematic. In order to capture the essence of the numerical convergence properties of our control derivative B-spline estimates, we restrict ourselves to a slight modification of the point-mass model of Section 3.1.2. Instead of setting $C_S = C_L$, we let, for ease of analysis,

$$C_S(u) = 2\pi u,$$

where 2π is the lift slope obtained from classical thin airfoil theory [1]. Initializing all B-spline estimator weights to 2π we would expect the weights to remain constant. As shown in Figure 5.9, however, the weight estimates slowly drift away from their original values.

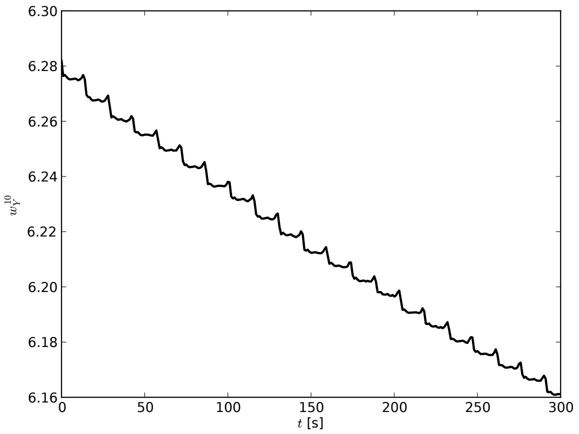

Figure 5.9: Drift in the estimate of the 10th Y-axis B-spline weight.

Such drift of correct parameter estimates – due to discretization, integration and measurement error – is known to occur when the input signal is not sufficiently rich [25, 36]. Several remedies have been proposed [25]. We briefly review the three most common ones.

Dead zones. The idea is to update estimates only when the tracking error exceeds certain bounds, i.e, modifying the modified estimator update law (4.16) to read

$$\frac{d\hat{\mathbf{w}}_i}{dt} = \begin{cases} \Gamma e_m P \lambda_i \tilde{u} \mathbf{b} & \text{if } |e_m| > D \\ 0 & \text{otherwise,} \end{cases}$$

where $D \in \mathbb{R}^+$ is the width of the dead zone.

Projection methods. The idea of projection is to constrain the coordinates of $\hat{\mathbf{w}}_i$ to lie in the closed interval $[-M, M]$. This can be achieved by modifying the estimator update law (4.16) to read

$$\frac{d(\hat{\mathbf{w}}_i)_j}{dt} = \begin{cases} \Gamma e_m P \lambda_i \tilde{u} \mathbf{b}_j & \text{if } |(\hat{\mathbf{w}}_i)_j| < M \\ & \text{or if } |(\hat{\mathbf{w}}_i)_j| = M \text{ and } e_m \lambda_i \tilde{u} (\hat{\mathbf{w}}_i)_j \mathbf{b}_j \leq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Leakage terms. The idea of the leakage term method is to modify the parameter estimate update laws so as to guarantee that – in face of bounded unmodeled dynamics – the parameter estimates converge to a given domain. The simplest of this family is the method of adding a *leakage term* of the form $-\Gamma L\hat{\mathbf{w}}_i$, $L \in \mathbb{R}^+$, to the estimator update law (4.16):

$$\frac{d\hat{\mathbf{w}}_i}{dt} = \Gamma e_m P \lambda_i \tilde{u}\mathbf{b} - \Gamma L\hat{\mathbf{w}}_i.$$

This is known as the σ-modification. The so-called e-modification adjusts the leakage term to be proportional to the tracking error. The details of these and other modifications are available in [25].

The control law (4.10) is robust against imperfect control derivative estimates in the sense that, running at 10 Hz, estimates of 0.5–2 times the real values still facilitate tracking. Note, however, that the resulting control input graph oscillates wildly; see 5.10 (compare Figure 5.4). Larger deviations from the real values of the control derivative B-spline weight estimates destroy the ability of the control loop to keep the kite in the air. These bounds imply that we could implement projection or leakage methods to constrain the estimates to the lie in the band of values that are 0.5–2 times nominal, but the observed parameter drift would then cause the estimates to converge to either boundary. This would defeat the purpose of our adaption process, i.e., reducing unnecessary control effort as much as possible.

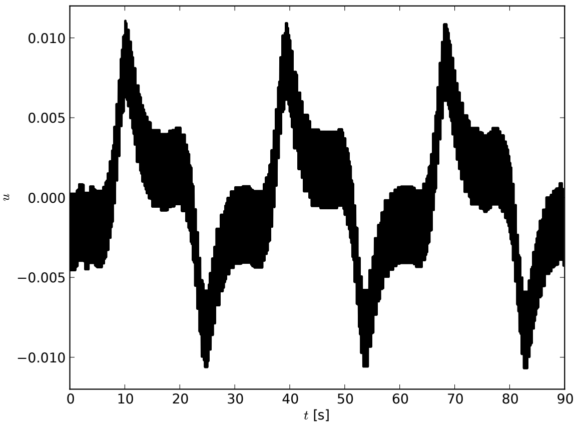

Figure 5.10: Steering input when the spline weights are set to 50% of their actual value.

As a consequence our preference extends to an application of dead-zone methods. Unfortunately, the robustness of our controller implies that tracking errors due to control derivative estimate errors are small. Numerical experiments confirm that introduction of a dead zone wide enough to eliminate parameter drift also filters out useful error information. Consequently, convergence of the parameter estimates is lost when introducing a dead zone; see Figure 5.11 for a comparison of convergence behaviour without and with a dead zone, respectively. Note how in the first graph the convergence becomes very slow after $t = 20$ s. The first 10 s of the simulation are not shown, as during this time adaptivity is switched off in an attempt to remove the influence of the initial tracking error on the weight estimates.

As a solution we apply a dead zone wide enough to exclude parameter drift, but disable the dead zone whenever the the rate of change of the controls exceeds a set maximum of 0.01 rad/s. In this way, adaption is turned on when the control input oscillates heavily (cf. Figure 5.10). The resulting convergence behaviour is plotted in Figure 5.11. Note that after $t = 20$ s the estimate has not converged to the true value, but it has become stationary. This contrasts with the scenario without a dead zone, where after $t = 20$ s convergence continues, albeit at a much slower pace. Around $t = 20$ s the values of the estimates are sufficiently good for the steering

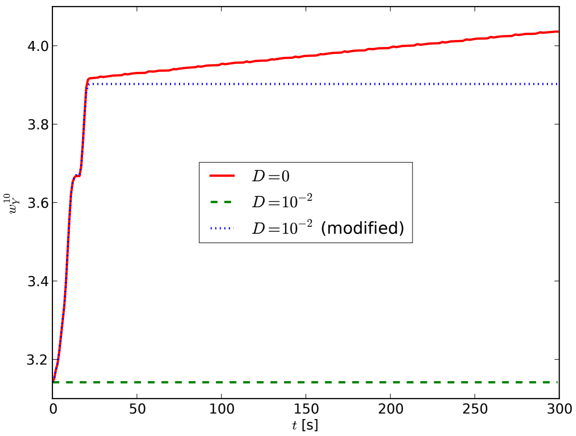

Figure 5.11: 10th Y-axis B-spline weight estimate ($C_S(u) = 2\pi u$, $\Gamma = 10^5$). Convergence without a dead zone (solid line), lack of convergence with dead zone (dashed line), and convergence with a modified dead zone (dotted line).

inputs not to exhibit major oscillations. We conclude that our *modified dead zone* renders the adaptivity of the controller useful for practical purposes.

Experiments with other initial estimates confirm the effectiveness of our modified dead zone modification. When setting C_S to the lift coefficient curve of Figure 3.4 and initializing the Y-axis control derivative B-spline weight estimates to π, we observe convergence behaviour very similar to the dotted line shown in Figure 5.11. Unfortunately, due to a bug in the state reduction procedure developed in [7] for the multi-body model (drift in the angles defining attitude of the kite), it is currently not possible to investigate convergence behaviour for the multi-body simulations.

To recapitulate, the modified dead zone approach comprises the following modifications:

1. Introduction of a dead zone to eliminate parameter drift;
2. Disabling the dead zone whenever the rate of change of the control input exceeds a given bound.

Both must be enabled simultaneously, for the dead zone alone destroys the convergence of the parameter estimates.

5.6 Comparison with linear control

For comparison we tuned a linear controller of the form

$$u = e \tag{5.3}$$

for the point-mass model. While for values such as $P = 0.03$ and $I = 0.09$ the linear control law (5.3) appears to achieve tracking, the disadvantage of the linearity is evident in the excessive control inputs applied at points where the steering direction changes; see Figure 5.12.

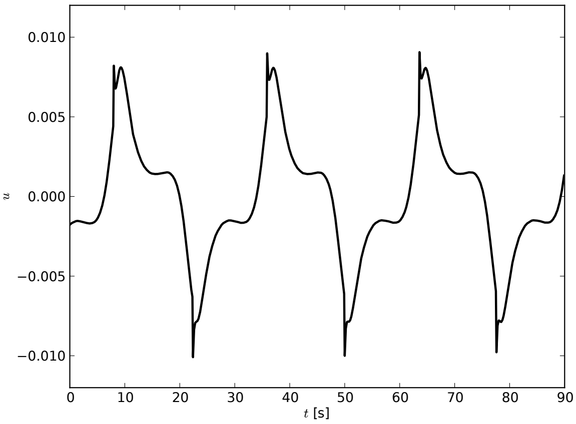

Figure 5.12: Steering input for linearly controlled kite model.

As before the controller runs at 10 Hz, the steering rate is limited to 0.05 rad/s, and the magnitude of the steering input to 0.015 rad.

Similarly to the non-linear control law (4.10), the linear controller (5.3) is robust in face of simplified Dryden [4] turbulence; see Figure 5.13. The mean linear tracking error (5.2) is, however, worse for most of the investigated range of turbulence intensities.

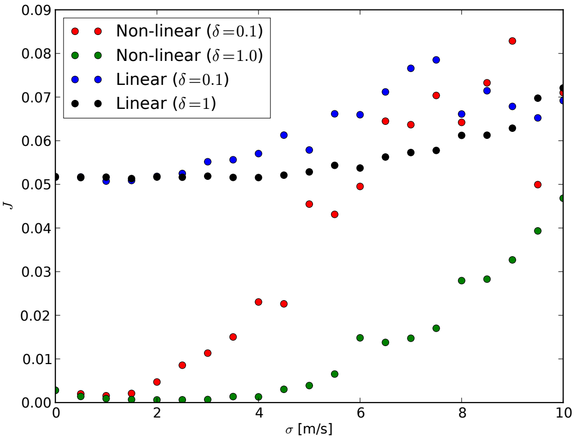

Figure 5.13: Mean linear tracking error as function of turbulence intensity σ and correlation rate δ, where $\sigma_{\text{lo}} = \sigma_{\text{la}} = \sigma_{\text{ve}} = \sigma$.

It is also worth noting that for $P = 0.15$ and $I = 0.45$ the linear control law is successful at steering the multi-body model.

We conclude that the linear control law performs surprisingly well. Less surprisingly, the non-linear controller outperforms the linear control law on all fronts (except for 4 data points at very high levels of turbulence).

5.7 Notes

The coupling of the controller implementation with the multi-body model, as discussed in Sections 5.1 and 5.4, was carried out together with Stefan de Groot.

Chapter 6

Outlook and future research

> *"The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work."*
>
> – John von Neumann.

6.1 Summary

In this thesis we presented a novel solution to the kite trajectory tracking problem using an explicit control law. Compared to alternative approaches, such as model predictive control, our approach has three major advantages: a stability proof, ease of implementation, and minimal modeling requirements. The latter is especially important for control of flexible kites, which are hard to model accurately in a point-mass or rigid-body framework.

Kites commonly have a single control input available for steering. We have shown how the differential-geometric notion of turning angle can be used as a one-dimensional representation of the kite trajectory, and how this leads to a single-input single-output tracking problem. In order to facilitate model inversion we linearized the turning angle dynamics in the steering control input, and applied energy methods to derive a stabilizing feedback law. We showed how the zero-term of the linearization can be measured directly using on-board sensors, and how in this way the control law comes to depend on the control derivatives of the aerodynamic kite model only. The controller adapts the estimates of these control derivatives based on tracking performance.

Repeated simulations with a point-mass model have shown our control approach to be robust against turbulence, and simulations with a multi-body model of a flexible kite have validated our modeling assumptions.

Discretization, integration and measurement errors induce subtle drift in the estimates of the control derivatives. In order to eliminate this drift we added a dead zone to the estimator update law. Unfortunately, the dead zone also filters out useful error information. As a solution, the dead zone is disabled whenever there are indications that the control derivatives have not converged to suitable values.

6.2 Recommendations for future research

Simulations show a feedback loop latency above 100 ms to seriously degrade control performance. This was confirmed by an unsuccesful initial test flight with the Laddermill prototype plant, which is currently not geared towards low-latency operation. If it is not possible to drive system latency down sufficiently, future research should focus on adding short-horizon prediction to the controller.

Another interesting development would be an outer-loop path planning controller to facilitate smooth transitions between different regimes of operation. A related problem is the design of suitable, possibly open-loop stable, target trajectories. The group led by Diehl at K.U. Leuven has made significant progress in this direction.

A third open problem is fully automated control of a pumping kite power plant such as the Laddermill prototype. Missing is a controller to automate the reeling in and out of the tether. The winch might perhaps be controlled together with the power setting of the kite, in order to target an appropriate line force. Coordination with a trajectory tracking controller might then be handled by a higher-level control system.

Bibliography

[1] J. D. Anderson. *Fundamentals of Aerodynamics.* McGraw-Hill Higher Education, 3rd edition, 2001.

[2] C. L. Archer and K. Caldeira. Global assessment of high-altitude wind power. *Energies*, 2(2):307–319, 2009.

[3] K. J. Astrom and B. Wittenmark. *Adaptive Control.* Addison-Wesley Longman Publishing Co., Inc., 2nd edition, 1994.

[4] T. R. Beal. Digital simulation of atmospheric turbulence for Dryden and von Karman models. *Journal of Guidance, Control, and Dynamics*, 16(1):132–138, 1993.

[5] J. Breukels. *An Engineering Methodology for Kite Design.* PhD thesis, Delft University of Technology, Delft, 2011.

[6] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi. *Wind Energy Handbook.* Wiley-Blackwell, 2001.

[7] S. G. C. de Groot, J. Breukels, R. Schmehl, and W. J. Ockels. Multibody system reduction for derivation of a flight dynamics model for real-time control of kites. Submitted to AIAA Journal of Guidance, Control and Dynamics, 2010.

[8] M. M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes.* PhD thesis, Ruprecht-Karls-Universität, Heidelberg, 2001.

[9] M. M. Diehl, R. Findeisen, F. Allgöwer, H. G. Bock, and J. P. Schlöder. Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEE Proc.-Control Theory Appl.*, 152(3):296–308, 2005.

[10] M. M. Diehl, L. Magni, and G. De Nicolao. Efficient NMPC of unstable periodic systems using approximate infinite horizon closed loop costing. *Annual Reviews in Control*, 28:37–45, 2004.

[11] J. F. Epperson. *An Introduction to Numerical Methods and Analysis.* John Wiley & Sons, 1st edition, 2002.

[12] L. Fagiano. *Control of Tethered Airfoils for High–Altitude Wind Energy Generation.* PhD thesis, Politecnico di Torino, Torino, 2009.

[13] J. Farrell, M. Polycarpou, and M. Sharma. Adaptive backstepping with magnitude, rate, and bandwidth constraints: Aircraft longitude control. In *American Control Conference*, volume 5, pages 3898–3904, Denver, Colorado, 2003.

[14] A. Furey and I. Harvey. Evolution of neural networks for active control of tethered airfoils. In *Proceedings of the 9th European Conference on Artificial Life*, pages 746–756. Springer-Verlag, 2007.

[15] A. Furey and I. Harvey. Adaptive behavioural modulation and hysteresis in an analogue of a kite control task. In *Simulation of Adaptive Behaviour*, 2008.

[16] A. Furey and I. Harvey. Robust adaptive control for kite wind energy using evolutionary robotics. *Biological Approaches to Engineering*, pages 117–120, 2008.

[17] A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC-Press, 2nd edition, 1997.

[18] G. Guennebaud, B. Jacob, et al. Eigen v2. `http://eigen.tuxfamily.org`, 2011.

[19] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, 31(3):363–396, 2005.

[20] B. Houska. Robustness and stability optimization of open-loop controlled power generating kites. Master's thesis, Ruprecht-Karls-Universität, Heidelberg, 2007.

[21] B. Houska and M. M. Diehl. Optimal control for power generating kites. In *9th European Control Conference*, pages 3560–3567, Kos, Greece, 2007.

[22] B. Houska and M. M. Diehl. Robustness and stability optimization of power generating kite systems in a periodic pumping mode. In *IEEE Multi - Conference on Systems and Control*, Yokohama, Japan, 2010.

[23] O. Härkegård and S. T. Glad. *Flight control design using backstepping*. Division of Automatic Control, Department of Electrical Engineering, Linköpings universitet, 2001.

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

[25] P. A. Ioannou and J. Sun. *Robust Adaptive Control*. Prentice Hall, 1996.

[26] T. P. Jung. Wind tunnel study of drag of various rope designs. In *28th AIAA Applied Aerodynamics Conference*, San Antonio, Texas, 2009.

[27] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, 2001.

[28] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic. *Nonlinear and adaptive control design*. John Wiley & Sons, New York, 1995.

[29] J. P. LaSalle. Stability theory for ordinary differential equations. *Journal of Differential Equations*, 4(1):57–65, 1968.

[30] M. L. Loyd. Crosswind kite power. *Journal of Energy*, 4(3):106–111, 1980.

[31] D. J. C. MacKay. *Sustainable Energy – without the hot air*. UIT, Cambridge, United Kingdom, 2008.

[32] J. P. Meijaard, W. J. Ockels, and A. L. Schwab. Modelling of the dynamic behaviour of a Laddermill, a novel concept to exploit wind energy. In *3rd International Symposium on Cable Dynamics*, Trondheim, Norway, 1999.

[33] W. J. Ockels. Laddermill, a novel concept to exploit the energy in the airspace. *Aircraft Design*, 4(2-3):81–97, 2001.

[34] R. E. Sheldahl and P. C. Klimas. Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines. Technical report, Sandia National Laboratories, Albuquerque, NM, 1981.

[35] S. Sieberling, Q. P. Chu, and J. A. Mulder. Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. *Journal of Guidance, Control, and Dynamics*, 33(6):1732–1742, 2010.

[36] L. Sonneveldt. *Adaptive Backstepping Flight Control for Modern Fighter Aircraft.* PhD thesis, Delft University of Technology, Delft, 2010.

[37] L. Sonneveldt, Q. P. Chu, and J. A. Mulder. Nonlinear flight control design using constrained adaptive backstepping. *Journal of Guidance, Control, and Dynamics*, 30(2):322–336, 2007.

[38] P. Williams, B. Lansdorp, and W. J. Ockels. Flexible tethered kite with moveable attachment points, part I: Dynamics and control. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Hilton Head, South Carolina, 2007.

[39] P. Williams, B. Lansdorp, and W. J. Ockels. Nonlinear control and estimation of a tethered kite in changing wind conditions. *Journal of Guidance, Control and Dynamics*, 31(3):793–799, 2008.

Appendix: Facts from differential geometry

In this section we list the main results from differential geometry referred to in this paper, adapted from the book by Gray [17]. We will not make the notions of surface and curve precise here, but only note that a *coordinate patch* is a differentiable map between an open set $\mathcal{U} \subset \mathbb{R}^2$ and $\mathbb{R}^3$: the intuition is that it associates coordinates to a part of a surface.

In the following, J will denote a counterclockwise rotation by $\pi/2$ in the tangent plane of the *oriented* surface under consideration.

We start with the formal definition of the turning angle. Intuitively the turning angle is the angle between the tangent of a curve and a reference, without being restricted to $[-\pi, \pi]$:

Definition 6.2.1. *Let $\mathcal{M}$ be an oriented surface, suppose that $\boldsymbol{\alpha} : (a,b) \to \mathcal{M}$ is a regular curve, and that $\mathbf{X}$ is an* everywhere nonzero *vector field along $\boldsymbol{\alpha}$. Fix t_0 with $a < t_0 < b$. Let θ_0 be a number such that*

$$\frac{\dot{\boldsymbol{\alpha}}(t_0)}{\|\dot{\boldsymbol{\alpha}}(t_0)\|} = \cos\theta_0 \frac{\mathbf{X}(t_0)}{\|\mathbf{X}(t_0)\|} + \sin\theta_0 \frac{J\mathbf{X}(t_0)}{\|\mathbf{X}(t_0)\|}.$$

Then there exists a unique differentiable function $\theta = \theta[\boldsymbol{\alpha}, \mathbf{X}] : (a,b) \to \mathbb{R}$ such that $\theta(t_0) = \theta_0$ and

$$\frac{\dot{\boldsymbol{\alpha}}}{\|\dot{\boldsymbol{\alpha}}\|} = \cos\theta[\boldsymbol{\alpha}, \mathbf{X}] \frac{\mathbf{X}}{\|\mathbf{X}\|} + \sin\theta[\boldsymbol{\alpha}, \mathbf{X}] \frac{J\mathbf{X}}{\|\mathbf{X}\|},$$

at all points on the curve. We call $\theta[\boldsymbol{\alpha}, \mathbf{X}]$ the turning angle *of $\boldsymbol{\alpha}$ with respect to $\mathbf{X}$ determined by θ_0 and t_0.*

In order to state Liouville's theorem, which allows us to compute the turning angle of a curve, we need the notion of *geodesic curvature.* Intuitively speaking the geodesic curvature of a curve measures how far it is from being a geodesic, that is, the shortest path between two points on the surface:

Definition 6.2.2. *Let $\boldsymbol{\alpha} : (a,b) \to \mathcal{M}$ be a curve in an oriented regular surface $\mathcal{M}$ in $\mathbb{R}^3$. Then the* geodesic curvature *of $\boldsymbol{\alpha}$ is defined (for $a < t < b$) as*

$$\kappa_g[\boldsymbol{\alpha}](t) = \frac{\ddot{\boldsymbol{\alpha}}(t) \cdot J\dot{\boldsymbol{\alpha}}(t)}{\|\dot{\boldsymbol{\alpha}}(t)\|^3}.$$

Finally, we state Liouville's theorem, which relates the geodesic curvature of a curve to the geodesic curvatures of the coordinate lines induced by the coordinate patch and the turning angle of the curve. Let $(\kappa_g)_1$ and $(\kappa_g)_2$ denote the geodesic curvatures of the coordinate lines $u \mapsto \mathbf{x}(u,v)$ and $v \mapsto \mathbf{x}(u,v)$, where $\mathbf{x}$ is a given patch:

Theorem 6.2.1 (Liouville)**.** *Let $\mathcal{M}$ be an oriented surface, and suppose that $\alpha : (a, b) \to \mathcal{M}$ is a regular curve whose trace is contained in $\mathbf{x}(\mathcal{U})$, where $\mathbf{x} : \mathcal{U} \to \mathcal{M}$ is a coordinate patch such that $\mathbf{x}_v = J\mathbf{x}_u$. Then*

$$\kappa_g[\boldsymbol{\alpha}] = (\kappa_g)_1 \cos\theta + (\kappa_g)_2 \sin\theta + \frac{\dot{\theta}}{\|\dot{\boldsymbol{\alpha}}\|},$$

where $\theta = \theta[\boldsymbol{\alpha}, \mathbf{x}_u]$ is the angle between $\boldsymbol{\alpha}$ and $\mathbf{x}_u$.