Power-optimal Operational Trajectories for Airborne Wind Energy

Titouan Renard [1], supervised by Johannes Waibel and Pr. Colin Jones

[1] MT-RO, 272257

Abstract *The following report describes an approach for the generation of power-optimal trajectories for a lift-mode rigid-kite airborne wind energy system. First it details a model of the system then covers it's transcription to a nonlinear program using direct collocation techniques, finally an implementation of a homotopy technique which enables more robust convergence of the optimization is described.*

1 Introduction

1.1 Context

Airborne wind energy (AWE) is an emerging field in renewable power which intends to harvest wind energy using airborne devices. They do so by exploiting the relative velocity between an airmass (the wind) and the ground to create a traction forces that can in turn be converted into electrical power.

Three major arguments are often listed to motivate the development of AWE systems [7]:

1. Like solar power, wind power provides enough energy to cover all of humanity's energy need.
2. AWE systems can reach higher altitudes than ground-based wind turbines, which enables them to tap into otherwise unusable energy resources.
3. AWE might need less material investment per unit of outputted power compared to ground-based wind energy.

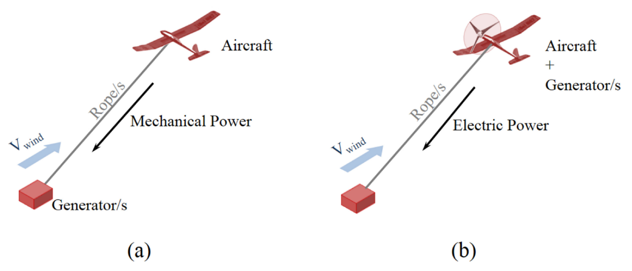

Fig. 1: Drag mode and lift mode AWE systems. [10]

There are two main AWE power transmission concepts: lift mode and drag mode (see figure 2). In drag mode, an airborne turbine is used to generate electrical power in the air, that electricl power is then transmitted to the ground through the tether. In lift mode a kite is flying in crosswind patterns while pulling on a tether which drives a generator on the ground.

Lift mode AWE concepts are further separated into rigid kite concepts, where the "kite" has a design which is close to a conventional airplane and has onboard control surfaces, and soft kite concepts where the kite is closer to a hobby kite. In this project we study a rigid kite, lift mode AWE system, more specifically we work with a model of the prototype small scale AWE system described in [16]. Several companies are developing comparable lift-mode systems: the Dutch *Ampyx Power* [17], the Swiss *TwingTec* [19] and the Norwegian *Kitemill* [18] are notable examples.

1.2 Problem definition

The problem we tackle in the subsequent report is the following, we aim to develop robust methods for the generation of power-optimal (in the sense that flying them yields a maximum average generated power) trajectories for a rigid wing, glider-like, lift mode AWE system (as schematically represented in figure 2). The system can be roughly separated in three component: a kite, a tether and a winch. The kite that we work with is an off the shelf RC glider fitted with custom control hardware and with an attachment mechanism for the tether (see [16]).

1.3 Literature review

Trajectory optimization and model predictive control AWE systems, are made especially difficult by the fact that key simplifications that are often made in conventional flight control are irrelevant for a tethered system. The wind cannot be neglected, and the spherical shape that the tether imposes implies the flight path is continuously changing, i.e. there is no stationary flight regime. Those properties make this kind of systems particularly well suited for Nonlinear Model Predictive Control [9].

The subject of this project is trajectory optimization rather than control, more specifically we make use of collocation techniques to transcribe an optimal control problems into non-linear programs. A good primer on those techniques can be found in [13].

Several publications make an argument for the use of trajectory optimization techniques for AWE, [4] presents an NMPC controller of a simplified model of an AWE kite, [8] discusses a relaxation scheme for dealing with the non-linearities of a lift-mode AWE trajectory optimization problem. A case study with experimental data from the *Ampyx* AWE system is discussed in [15].

2 Methods

In the following section, we describe a dynamical model of the AWE system (section 2.1), then we describe an optimal control problem formulation for the generation of power-optimal trajectories (section 2.2) and describe the transcription of that OCP into a nonlinear program (section 2.2.4), finally, we discuss a homotopy techniques that allows us to solve otherwise extremely hard NLPs (section 2.2.5).

The model is implemented in *Matlab* using the algorithmic differentiation framework *Casadi* [14]. The optimal control problem is formulated within the *Casadi* framework and then is solved using the NLP solver *IPOPT* [3]. The *Matlab* code implementing the methods described in the following section and which can be used to reproduce the results discussed in section 3 is submitted together with this report.

2.1 System Model

We describe a system of differential algebraic equations that provide a dynamical model for the AWE system. We discuss two level of fidelity of the model: a *full model* with rotational dynamics and aerodynamic moments and a simplified *kinematic model*.

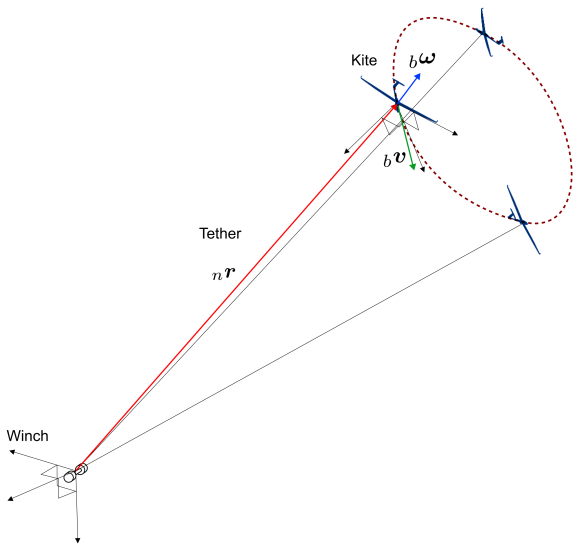

Fig. 2: Overview of the system's components, most important reference frames and state vectors $_n\boldsymbol{r}$, $_b\boldsymbol{v}$ and $_b\boldsymbol{\omega}$.

2.1.1 Reference frames and attitude representation: For the purpose of the dynamic model, we use two reference frames. The *ground reference frame*, which is attached to a fixed point on the ground and the *body reference frame*, which is attached to the body of the kite (as in figure 3). The ground frame uses NED coordinates [12]. We neglect the effect of earth's rotation and work under a flat earth hypothesis.

We choose to express orientation and rotations using unit quaternions (sometimes also referred to as versors). Quaternions give a representation of the SO(3) rotation group. They are commonly used in control, robotics and computer graphics because of their numerical stability, their compactness (compared to rotation matrices) and the fact that they are not subject to gimbal lock (unlike Euler angles). As in [16] we use the $\cdot$ (dot) operator to denote the transformation of a vector by a transformation quaternion:

$$\boldsymbol{q}_{\text{rot}} \cdot \boldsymbol{v} := \boldsymbol{q}_{\text{rot}} \otimes \begin{bmatrix} 0 \\ \boldsymbol{v} \end{bmatrix} \otimes \boldsymbol{q}_{\text{rot}}^{-1}. \tag{1}$$

One important property of unit quaternions as a representation of the 3d rotation group SO(3) is that they provide a double-cover of the group. In practical terms this means that two different transformation quaternions map to each unique rotation in SO(3). Specifically, q and $-q$ represent the same rotation. This causes problems when comparing rotations through quaternions, we overcome this by defining an error quaternion for rotation:

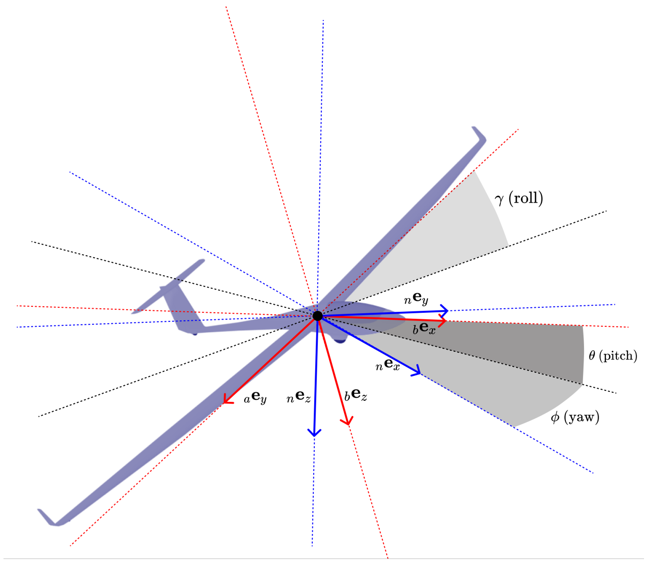

Fig. 3: Ground and body reference frames.

$$\boldsymbol{q}_{\text{error}} = \boldsymbol{q}_a^{-1} \otimes \boldsymbol{q}_b. \tag{2}$$

The error quaternions defined in (eq. 2) has the following property:

Property 1. *$\boldsymbol{q}_a^{-1} \otimes \boldsymbol{q}_b = 1 + 0i + 0j + 0k \iff \boldsymbol{q}_a$ and $\boldsymbol{q}_b$ represent the same rotation in SO(3).*

This allows us to define a distance metric between two rotation as:

Definition 1. *Let $\Delta(\boldsymbol{q}_a, \boldsymbol{q}_b) := \|\boldsymbol{q}_a^{-1} \otimes \boldsymbol{q}_b - 1\|$ be a distance metric between two rotations a and b represented by quaternions $\boldsymbol{q}_a$ and $\boldsymbol{q}_b$. Here $\|\cdot\|$ denotes the quaternion norm and the 1 subtracted to the error quaternion is a quaternion itself ($1 = 1 + 0i + 0j + 0k$).*

We will use that metric to impose constraints on our dynamic model.

In the context of our dynamics model, let us define two useful quaternions describing rotations between both our reference frames. We call $\boldsymbol{q}_{nb}$ the quaternion describing the rotation from the ground NED frame to the body frame and $\boldsymbol{q}_{bn} = \boldsymbol{q}_{nb}^{-1}$ the quaternion representing rotation from the body to the ground frame.

The derivative of a quaternion is related to angular velocities in Euler angles by the following equality [5]:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{q}_{\boldsymbol{\omega}}. \tag{3}$$

Where $\boldsymbol{q}_{\boldsymbol{\omega}}$ can be computed from $\boldsymbol{\omega} \in \mathbb{R}^3$ as follows:

$$\boldsymbol{q}_{\omega} = 0 + i\omega_x + j\omega_y + k\omega_z. \tag{4}$$

The derivative calculated in eq. 3 is correct but tends to lead to norms of quaternions drifting away from 1 when integrated numerically. To overcome that problem we use a stabilization method inspired by [11]:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{q}_{\boldsymbol{\omega}} + \frac{1}{2}\lambda \boldsymbol{q}_{nb}(\|\boldsymbol{q}_{nb}\|^2 - 1). \tag{5}$$

2.1.2 State and input variables: The full model uses $\boldsymbol{x} \in \mathbb{R}^{14}$ as a state vector:

$$\boldsymbol{x} = \begin{bmatrix} {}_b\boldsymbol{v} \\ {}_b\boldsymbol{\omega} \\ {}_n\boldsymbol{r} \\ \boldsymbol{q}_{nb} \\ l \end{bmatrix}. \tag{6}$$

Where ${}_b\boldsymbol{v} = ({}_bv_x, {}_bv_y, {}_bv_z)^T$ denotes the plane's velocity in the body frame, ${}_b\boldsymbol{\omega} = ({}_b\omega_x, {}_b\omega_y, {}_b\omega_z)^T$ the plane's angular velocity in the body frame, ${}_n\boldsymbol{r} = ({}_nr_x, {}_nr_y, {}_nr_z)^T$ the plane's position in the ground frame. $\boldsymbol{q}_{nb}$ is the quaternion defining the rotation from the ground frame to the body frame, finally, l is the relaxed length of the tether (assuming no elongation).

By convention the tether is attached to the $(0, 0, 0)$ point of the ground frame, which implies the ${}_n\boldsymbol{r}$ vector gives a description of exact distance between the winch and the kite, which can in turn be used to compute the tether's elongation (see section 2.1.4).

The input vector $\boldsymbol{u} \in \mathbb{R}^5$ is defined as follows:

$$\boldsymbol{u} = \begin{bmatrix} dE \\ dR \\ dA \\ \tau \\ dl \end{bmatrix}, \tag{7}$$

where $(dE, dR, dA)^T = \boldsymbol{u}_D$ describes control surfaces deflection for elevons, rudder and ailerons, τ is the engine thrust and dl is the tether reel-in/out speed.

In the case of the kinematic model, we directly use the angular velocity as input which yields a reduced dimensionality state $\boldsymbol{x}_{\text{kin}} \in \mathbb{R}^1 1$ and an input $\boldsymbol{u}_{\text{kin}} \in \mathbb{R}^5$ as described below:

$$\boldsymbol{x} = \left[{}_b\boldsymbol{v}, {}_n\boldsymbol{r}, \boldsymbol{q}_{nb}, l\right]^T, \tag{8}$$

$$\boldsymbol{u} = \left[{}_b\boldsymbol{\omega}, \tau, dl\right]^T. \tag{9}$$

2.1.3 Aerodynamic forces and moments: The following section describes the model used to compute aerodynamic forces and moments acting on the aircraft in a given state. This model is taken from [16]. Let ${}_b\boldsymbol{v}$ be the speed vector of the aircraft (in body frame), ${}_n\boldsymbol{v}_W$ be the wind vector (in ground frame). This is illustrated in figure 4. We denote the airspeed vector as:

$${}_b\boldsymbol{v}_a = {}_b\boldsymbol{v} - \boldsymbol{q}_{bn} \cdot {}_n\boldsymbol{v}_W. \tag{10}$$

We then compute the the main aerodynamic variables from ${}_b\boldsymbol{v}_a$:

Airspeed	$V_a = \lVert {}_b\boldsymbol{v}_a \rVert,$	(11)
Angle of attack	$\alpha = \arctan\left(\frac{{}_b\boldsymbol{v}_{a,z}}{{}_b\boldsymbol{v}_{a,x}}\right),$	(12)
Sideslip angle	$\beta = \arcsin n\left(\frac{{}_b\boldsymbol{v}_{a,y}}{V_a}\right),$	(13)
Dynamic pressure	$\bar{q} = \frac{\rho}{2V_a^2}.$	(14)

The lift coefficient C_L, drag coefficient C_D, sideslip coefficient C_S, the roll, pitch and yaw moment coefficient C_l, C_m, C_n are computed as follows:

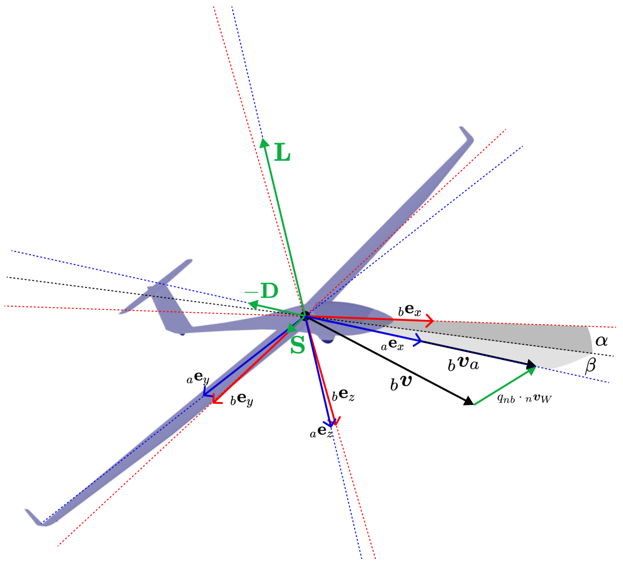

Fig. 4: Aerodynamic and body reference frames. Notice that these two reference frames are distinct: the body reference frame has it's x component aligned with the airplane's nose while the aerodynamic reference frame has it's x component aligned with the airspeed vector ${}_a\boldsymbol{v}$. and that the components $-D$, S and L of the aerodynamic force are aligned on the aerodynamic reference frame.

$$\mathrm{C}_L = \mathrm{CL}_0 + \mathrm{CL}_\alpha \alpha + \mathrm{CL}_q \frac{c \cdot \omega_x}{2V_a} + \mathrm{CL}_{de} \cdot \mathrm{dE}, \tag{15}$$

$$\mathrm{C}_D = \mathrm{CD}_0 + \mathrm{CL}^2/(\pi e A_W), \tag{16}$$

$$\mathrm{C}_S = \beta \mathrm{CY}_b + \frac{b}{2V_a}(\mathrm{CY}_p \cdot \omega_x + \mathrm{CY}_r \omega_z) + \mathrm{CY}_{dr}\mathrm{dR}, \tag{17}$$

$$\mathrm{C}_l = \mathrm{Cl}_0 + \mathrm{Cl}_b \cdot \beta + \frac{b}{2 \cdot V_0} \cdot (\mathrm{Cl}_p \cdot \omega_x + \mathrm{Cl}_r \cdot \omega_z), \tag{18}$$

$$\mathrm{C}_m = \mathrm{Cm}_0 + \mathrm{Cm}_a \cdot \alpha + \frac{c}{2 \cdot V0} \cdot \mathrm{Cm}_q \cdot \omega_y + \mathrm{Cm}_{\mathrm{de}} \cdot \mathrm{dE}, \tag{19}$$

$$\begin{aligned} \mathrm{C}_n = \mathrm{Cn}_0 + \mathrm{Cn}_b \cdot \beta + \frac{b}{2 \cdot V_0} \cdot (\mathrm{Cn}_p \cdot \omega_x + \\ \mathrm{Cn}_r \cdot \omega_z) + \mathrm{Cn}_{\mathrm{da}} \cdot \mathrm{dA} + \mathrm{Cn}_{\mathrm{dr}} \cdot \mathrm{dR}. \end{aligned} \tag{20}$$

Where A_W is the wing surface, e is the Oswald efficiency coefficient and dE the deflection of the elevons. Finally the components of the aerodynamic forces are computed as:

$$\mathrm{L} = \bar{q} \cdot A_W \cdot \mathrm{C}_L, \tag{21}$$

$$\mathrm{D} = \bar{q} \cdot A_W \cdot \mathrm{C}_D, \tag{22}$$

$$\mathrm{S} = \bar{q} \cdot A_W \cdot \mathrm{C}_S. \tag{23}$$

The components of the aerodynamic moments are computed as:

$$l = \bar{q} \cdot A_W \cdot b \cdot \mathrm{C}_l, \tag{24}$$

$$m = \bar{q} \cdot A_W \cdot c \cdot \mathrm{C}_m, \tag{25}$$

$$n = \bar{q} \cdot A_W \cdot b \cdot \mathrm{C}_n. \tag{26}$$

Which give us an aerodynamic force ${}_a\boldsymbol{F}_A = [-\mathrm{D}, \mathrm{S}, \text{-L}]$ and moment ${}_b\boldsymbol{M}_A = [l, m, n]$ in the aerodynamic frame (where the first component is aligned with the airspeed vector ${}_b\boldsymbol{v}_a$). Which we bring into the body reference frame as follows:

$$_b\boldsymbol{F}_A = \boldsymbol{q}_{ba} \cdot {_a\boldsymbol{F}_A}. \tag{27}$$

Where $\boldsymbol{q}_a$ represents the rotations from the aerodynamic reference frame to the body reference frame (see figure 4).

$$\boldsymbol{q}_{ba} = \boldsymbol{q}_2(\alpha) \otimes \boldsymbol{q}_3(-\beta), \tag{28}$$

where $\boldsymbol{q}_2$ and $\boldsymbol{q}_3$ are elementary rotation quaternions around the y and z axes.

2.1.4 Tether and winch dynamics: The tether is modeled as three independent force components: a longitudinal elongation component, an aerodynamic drag component and a weight component. We denote the full tether force as:

$$_b\boldsymbol{F}_T = {_b\boldsymbol{F}_{\text{lon}}} + {_b\boldsymbol{F}_{\text{drag}}} + {_b\boldsymbol{F}_{\text{weight}}}. \tag{29}$$

Since the tether is not attached to the center of mass of the plane, but rather is offset by a vector $_b\boldsymbol{r}_{\text{out}} = (x_{\text{out}}, 0, z_{\text{out}})^T$ it generates a moment that we compute as:

$$_b\boldsymbol{M}_T = {_b\boldsymbol{r}_{\text{out}}} \times {_b\boldsymbol{F}_T}. \tag{30}$$

The longitudinal component is computed in the direction of the tether's elongation (and is therefore collinear with $_n\boldsymbol{r}$), it is proportional to the elongation of the tether, which is computed as follows:

$$\epsilon_{\text{tether}} = \frac{\|_n\boldsymbol{r}\| - l}{l}. \tag{31}$$

Given the tether's young modulus E and it's cross area $A_T = \pi r_T^2$ we can compute it's spring constant $k = E \cdot A_T$ and use it to get the following expression for the longitudinal force's magnitude:

$$F_{\text{lon,disc}} = \max(0, k\epsilon) = (k\epsilon)_+. \tag{32}$$

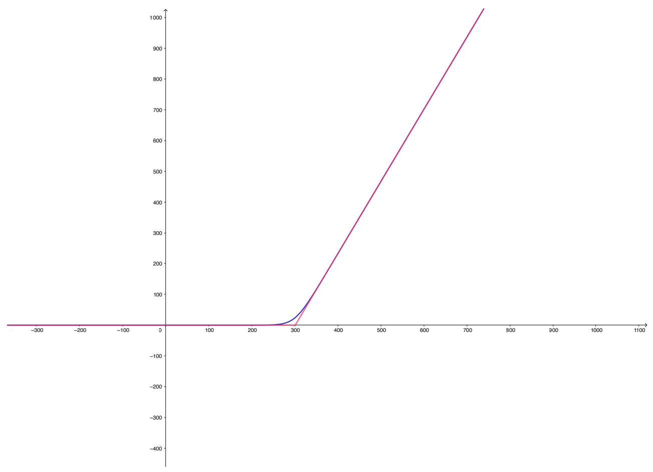

Fig. 5: The $\hat{p}(x)$ function (in blue) provides a smooth approximation of the $(\cdot)_+$ function (in red).

Since we will perform algorithmic differentiation on our model for optimal control, we require our function to have a continuous derivative. We choose, as suggested in [1], to use the following C^∞-smooth function to approximate the $(\cdot)_+$ function (see figure 5):

$$\hat{p}(x) = \frac{1}{\alpha} \log(1 + \exp(\alpha \cdot x)). \tag{33}$$

Where $\alpha > 0$ is a parameter such that $\forall x \in \mathbb{R}$:

$$\lim_{\alpha \to +\infty} \hat{p}(x) = (x)_+ \tag{34}$$

Which gives us the following approximate expression for the longitudinal tether force:

$$\|\boldsymbol{F}_{\text{lon}}\| = \frac{k}{l} \cdot \hat{p}(\|_b\boldsymbol{r}_{\text{out}}\| - l), \tag{35}$$

$$_b\boldsymbol{F}_{\text{lon}} = \boldsymbol{q}_{bn} \cdot \left[\frac{_n\boldsymbol{r}}{\|_n\boldsymbol{r}\|} \|\boldsymbol{F}_{\text{lon}}\| \right]. \tag{36}$$

The drag component is proportional to the apparent velocity orthogonal to the tether $_l\boldsymbol{v}_{A,\text{proj}}$ [6]:

$$_l\boldsymbol{v}_A = \boldsymbol{q}_{ln} \cdot {_n\boldsymbol{v}_a}, \tag{37}$$

$$_l\boldsymbol{v}_{A,\text{proj}} = \left[{_l\boldsymbol{v}_{Ax,y}}, 0 \right]^T. \tag{38}$$

We compute the drag component as follows:

$$_l\boldsymbol{D}_T = -\frac{1}{8}\rho \cdot {_l\boldsymbol{v}_{A,\text{proj}}} \cdot \|_l\boldsymbol{v}_{A,\text{proj}}\| \cdot c_\perp d_T l, \tag{39}$$

$$_b\boldsymbol{D}_T = \boldsymbol{q}_{bl} \cdot {_l\boldsymbol{D}_T}. \tag{40}$$

Where, ρ gives the air density, $c_\perp$ is the drag coefficient of the tether and d_T it's diameter.

The tether's weight component is given by the following equations:

$$m_T = \rho_T \cdot A_T \cdot l, \tag{41}$$

$$_b\boldsymbol{W}_T = \boldsymbol{q}_{bn} \cdot \left[0, 0, m_T \cdot g.\right]^T \tag{42}$$

Where ρ_T gives the tether's density, A_T it's cross section.

2.1.5 Translational Dynamics: The translational dynamics of the system is given (in the body frame) by Newton's second law of motion ($\frac{\partial p}{\partial t} = \sum F_e$):

$$m \cdot {_b\dot{\boldsymbol{v}}} = {_b\boldsymbol{F}_A} + {_b\boldsymbol{F}_T} + {_b\boldsymbol{F}_\tau} + {_b\boldsymbol{F}_W} - ({_b\boldsymbol{\omega}} \times {_b\boldsymbol{v}}), \tag{43}$$

$$_b\dot{\boldsymbol{v}} = \frac{1}{m} \sum {_b\boldsymbol{F}_E} - ({_b\boldsymbol{\omega}} \times {_b\boldsymbol{v}}). \tag{44}$$

Where $_b\boldsymbol{F}_A$ is the aerodynamic force described in section 2.1.3, $_b\boldsymbol{F}_T$ is the tether force described in section 2.1.4, $_b\boldsymbol{F}_\tau = [\tau, 0, 0]$ is the thrust force and $_b\boldsymbol{F}_W = \boldsymbol{q}_{bn} \cdot [0, 0, mg]$ is the plane's weight (see figure 6).

2.1.6 Rotational Dynamics: The rotational dynamics of the system is similarly written as:

$$\boldsymbol{I} \cdot {_b\dot{\boldsymbol{\omega}}} = {_b\boldsymbol{M}_A} + {_b\boldsymbol{M}_T} - {_b\dot{\boldsymbol{\omega}}} \times \boldsymbol{I} \cdot {_b\dot{\boldsymbol{\omega}}}, \tag{45}$$

$$_b\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1} \left(\sum {_b\boldsymbol{M}_E} - {_b\dot{\boldsymbol{\omega}}} \times \boldsymbol{I} \cdot {_b\dot{\boldsymbol{\omega}}} \right). \tag{46}$$

Where $\boldsymbol{I}$ contains the moment of inertia of the airframe:

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{47}$$

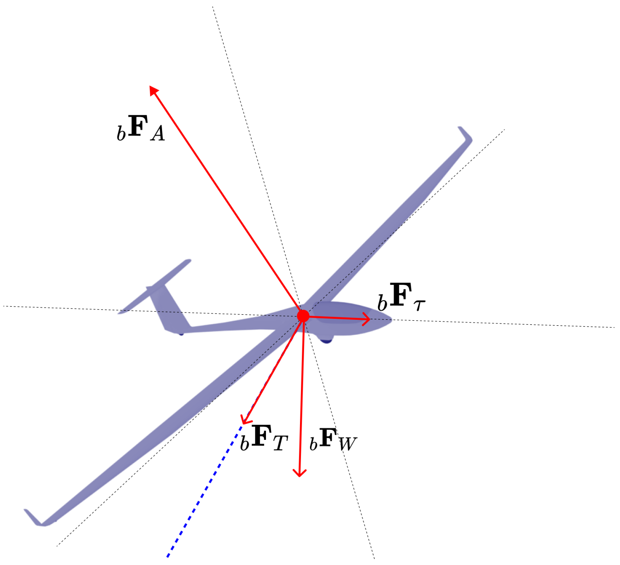

Fig. 6: Forces acting on the airframe. Notice the tether in blue.

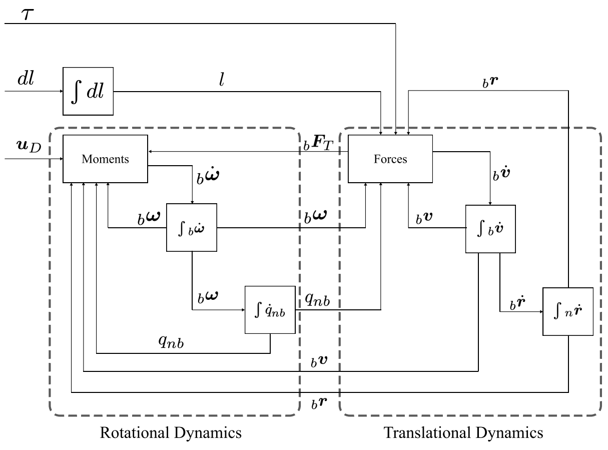

Fig. 7: Block diagram of the full dynamical model of the system.

2.1.7 Dynamic and Kinematic models: We consider two different levels of fidelity for the modeling of our system: a full model and a kinematic model. The differential equations are the formulated in

$$\dot{x} = f(\boldsymbol{x}, \boldsymbol{u})$$

form. In the case of the full model (see block diagram figure 7), $f(\boldsymbol{x}, \boldsymbol{u})$ is given by:

$$f(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} {_b\dot{\boldsymbol{v}}} \\ {_b\dot{\boldsymbol{\omega}}} \\ {_n\dot{\boldsymbol{r}}} \\ \dot{\boldsymbol{q}}_{bn} \\ \dot{l} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{m}\sum {_b\boldsymbol{F}_E} - ({_b\boldsymbol{\omega}} \times {_b\boldsymbol{v}}) \\ \boldsymbol{I}^{-1}(\sum {_b\boldsymbol{M}_E} - {_b\dot{\boldsymbol{\omega}}} \times \boldsymbol{I} \cdot {_b\dot{\boldsymbol{\omega}}}) \\ \boldsymbol{q}_{nb} \cdot {_b\boldsymbol{v}} \\ \frac{1}{2}\boldsymbol{q}_{bn} \otimes \boldsymbol{q}_\omega + \frac{1}{2}\lambda \boldsymbol{q}_{bn}(\|\boldsymbol{q}_{bn}\|^2 - 1) \\ dl \end{bmatrix}, \tag{48}$$

with ${_b\dot{\boldsymbol{v}}}$ as derived in section 2.1.5, ${_b\dot{\boldsymbol{\omega}}}$ as derived in section 2.1.6, ${_n\dot{\boldsymbol{r}}}$ simply uses the property $\frac{\partial {_b\boldsymbol{r}}}{\partial t} = {_b\boldsymbol{v}}$, $\dot{\boldsymbol{q}}_{bn}$ as derived in section 2.1.1 and $\dot{l} = dl$ by definition of dl.

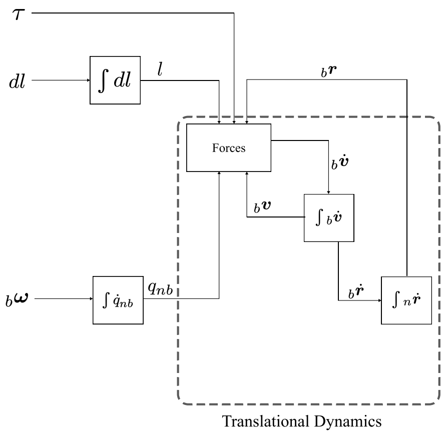

Fig. 8: Block diagram of the kinematic model of the system.

In the case of the kinematic model (see block diagram figure 8) the differential equations are quite similar but since we directly use ${_b\boldsymbol{\omega}}$ as input the rotational dynamics are replaced by a simple integrator (where angle is represented as a transformation quaternion):

$$f_{\text{kin}}(\boldsymbol{x}_{\text{kin}}, \boldsymbol{u}_{\text{kin}}) = \begin{bmatrix} {_b\dot{\boldsymbol{v}}} \\ {_n\dot{\boldsymbol{r}}} \\ \dot{\boldsymbol{q}}_{bn} \\ \dot{l} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{m}\sum {_b\boldsymbol{F}_E} - ({_b\boldsymbol{\omega}} \times {_b\boldsymbol{v}}) \\ \boldsymbol{q}_{nb} \cdot {_b\boldsymbol{v}} \\ \frac{1}{2}\boldsymbol{q}_{bn} \otimes \boldsymbol{q}_\omega + \frac{1}{2}\lambda \boldsymbol{q}_{bn}(\|\boldsymbol{q}_{bn}\|^2 - 1) \\ dl \end{bmatrix}. \tag{49}$$

When working with the kinematic model we assume that $(dE, dR, dA) = (0, 0, 0)$ for the computation of the aerodynamic forces (see section 2.1.3).

2.1.8 Power generation: Since this project focuses on optimizing the power generation and since $P_{\text{electrical}} \propto P_{\text{mechanical}}$ we directly work with mechanical power rather than modeling electromechanical power conversion. We estimate the average power $\hat{P}_{\text{mec}}$ output on a trajectory of period T as follows:

$$\hat{P}_{\text{mec}} = \frac{1}{T}\int_0^T \|\boldsymbol{F}_{\text{lon}}\| \cdot dl \cdot dt. \tag{50}$$

Where $\|\boldsymbol{F}_{\text{lon}}\|$ is the longitudinal component derived in section 2.1.4 eq. 35, and dl is the reel-in/out speed from the system's input.

2.2 OCP Formulation

Using the dynamical model described in section 2.1, one can formulate the AWE problem as an optimal control problem (OCP) in the following formulation:

$$\begin{aligned} \underset{\mathcal{X},\mathcal{U}}{\text{minimize}} \quad & \mathbb{L}(\mathcal{X},\mathcal{U}) && \text{(51a)} \\ \text{subject to} \quad & F(\dot{\mathcal{X}},\mathcal{X},\mathcal{U}) = 0, && \text{(51b)} \\ & h(\dot{\mathcal{X}},\mathcal{X},\mathcal{U}) \leq 0, && \text{(51c)} \\ & \Delta(\mathcal{X},\mathcal{U}) = 0. && \text{(51d)} \end{aligned}$$

with decision variables $\mathcal{X},\mathcal{U}$. The OCP minimizes the cost (51a) where $\mathbb{L}$ denotes the integrand cost, and is subject to a set of transcribed DAE (51b), as described in section 2.1, path constraints (51c) and periodicity constraints (51d).

2.2.1 Cost: The Lagrange cost (eq. 51a) is separated in three separate components as follows:

$$\mathbb{L}(\mathcal{X},\mathcal{U}) = \int_0^T \mathcal{L}(\boldsymbol{x}(t),\boldsymbol{u}(t))dt, \quad (52)$$

$$\mathcal{L}(\boldsymbol{x}(t),\boldsymbol{u}(t)) = C_{\text{path}} \cdot \epsilon_{\text{path}}(\boldsymbol{x}(t)) + C_{\text{control}}\epsilon_{\text{control}}(\boldsymbol{u}(t)) + C_{\text{phys}} \cdot \epsilon_{\text{phys}}(\boldsymbol{x}(t)) - C_{\text{mec}} \cdot P_{\text{mec}}(\boldsymbol{x}(t),\boldsymbol{u}(t)). \quad (53)$$

Where C_{path}, C_{control}, C_{phsy} and C_{mec} are linear weight coefficients, $\epsilon_{\text{path}}(\boldsymbol{x}(t))$ an error coefficients on a tracked path, $\epsilon_{\text{control}}(\boldsymbol{u}(t))$ a linear quadratic cost on control inputs, ϵ_{phys} an error coefficient encouraging physically sound behavior (for instance it can be used to penalized sideslip) and P_{mec} is the mechanical power computed in section 2.1.8.

The path tracking error ϵ_{path} is computed as a squared distance to a circle tilted around the winch by an angle θ_{tilt} (we represent that rotation by the quaternion q_{tilt}), we compute this error as follows:

$$_{\text{tilt}}\boldsymbol{r} = q_{\text{tilt}} \cdot {}_n\boldsymbol{r} = [_{\text{tilt}}\boldsymbol{r}_{x,y}, {}_{\text{tilt}}r_z]^T \in \mathbb{R}^3, \quad (54)$$

$$\boldsymbol{r}_{\text{virtual circle}} = R_{\text{path}} \frac{_{\text{tilt}}\boldsymbol{r}_{x,y}}{\|_{\text{tilt}}\boldsymbol{r}_{x,y}\|}, \quad (55)$$

$$\epsilon_{\text{path}} = \|\, \boldsymbol{r}_{\text{virtual circle}} - {}_{\text{tilt}}\boldsymbol{r}\|^2 + (z_{\text{path}} - {}_{\text{tilt}}r_z)^2. \quad (56)$$

Where R_{path}, the radius of the tracked circle and z_{path}, the tracked virtual altitude, are free parameters.

The control cost $\epsilon_{\text{control}}(\boldsymbol{u}(t))$ is given by the following equation:

$$\epsilon_{\text{control}}(\boldsymbol{u}) = \boldsymbol{u}^T R \boldsymbol{u}, \quad (57)$$

where R is a diagonal const matrix.

The physical cost component ϵ_{phys} is a quadratic penalty on sideslip and is computed as follows:

$$\epsilon_{\text{phys}} = (_b v_y)^2. \quad (58)$$

The mechanical power cost component approximates the average power derived in eq. 50, it is implemented as follows:

$$P_{\text{mec}} = \|\boldsymbol{F}_{\text{lon}}\| \cdot dl. \quad (59)$$

2.2.2 Path constraints: The path constraints (eq. 51a) are split between state constraints, input constraints, and trajectory constraints.

State and input constraints are used to ensure the state (eq. 60a) and input (eq. 60b) variables are contained in the domain of plausible values, they are formulated as follows:

$$\boldsymbol{x}_{\text{LB}} \le \boldsymbol{x}(t) \le \boldsymbol{x}_{\text{UB}}, \forall t, \quad (60a)$$

$$\boldsymbol{u}_{\text{LB}} \le \boldsymbol{u}(t) \le \boldsymbol{u}_{\text{UB}}, \forall t. \quad (60b)$$

Additionally our problem is subject to the following trajectory constraints:

$$_n\boldsymbol{r}(t_i) = {}_n\boldsymbol{r}_i,\ (t_i, {}_n\boldsymbol{r}_i) \in \mathcal{R}, \quad (61a)$$

$$_n\boldsymbol{v}(t_i) = {}_n\boldsymbol{v}_i,\ (t_i, {}_n\boldsymbol{v}_i) \in \mathcal{V}, \quad (61b)$$

$$|\epsilon_{\text{tether}}(t)| < \epsilon_{\max}, \forall t. \quad (61c)$$

Where eq. 61a imposes that the kite passes through a set of position $_n\boldsymbol{r}_i$ at specific times t_i, eq. 61b imposes specific velocities $_n\boldsymbol{v}_i$ at specific times t_i and eq. 61c imposes that the tether elongation is kept within a reasonable value $\epsilon_{\max}$. Constraint 61b is not applied for most instances of the problem, but is used when generating an initial guess (see section 2.2.6). Constraint 61a is only applied on one or two well chosen points when computing power optimal solutions. Ideally having an unconstrained path would give the NLP solver more leeway to get a good solution, but it was experimentally found that adding such constraints was helpful to get convergence.

2.2.3 Periodicity: The periodicity constraints (eq. 51d) are used to ensure that optimized trajectory is periodic. We define a closure error $\Delta(\mathcal{X},\mathcal{U})$ as follows:

$$\Delta(\mathcal{X},\mathcal{U}) = \sum_{0\le t\le T} \|_n\boldsymbol{r}(T) - {}_n\boldsymbol{r}(0)\|^2 + \|_b\boldsymbol{v}(T) - {}_b\boldsymbol{v}(0)\|^2 + \|_n\boldsymbol{\omega}(T) - {}_n\boldsymbol{\omega}(0)\|^2 + \|q_{nb}(T)^{-1} \otimes q_{nb}(0) - 1\|^2 + (l(T) - l(0))^2 + \|\boldsymbol{u}(T) - \boldsymbol{u}(0)\|^2, \quad (62a)$$

$$\Delta_{\text{kin}}(\mathcal{X},\mathcal{U}) = \sum_{0\le t\le T} \|_n\boldsymbol{r}(T) - {}_n\boldsymbol{r}(0)\|^2 + \|_b\boldsymbol{v}(T) - {}_b\boldsymbol{v}(0)\|^2 \|q_{nb}(T)^{-1} \otimes q_{nb}(0) - 1\|^2 + (l(T) - l(0))^2 + \|\boldsymbol{u}_{\text{kin}}(T) - \boldsymbol{u}_{\text{kin}}(0)\|^2. \quad (62b)$$

2.2.4 Transcription: Because of the long time-horizon ($\approx 15s$) required to solve the periodic optimization problem for time-optimal trajectories, shooting methods are unpractical. We thus use a direct multi-segment orthogonal collocation method to discretize the problem.

In practice most of our results were achieved with 2$^{\text{nd}}$ degree polynomial, and multiple segments, which is also sometimes referred to as Hermite-Simpson collocation [13].

The resulting transcription is thereupon solved using sequential quadratic programming using the non linear program solver *IPOPT* [3], and the sparse linear solver *MUMPS* [2] for the underlying linear programs.

2.2.5 Homotopy Procedure: The problem stated in eq. 51 transcribed as in section 2.2.4 yields a non-linear program whose solutions correspond to a physically meaningful result. However, in practice, because of the strong non-linearity of the problem most initializations of the solver lead to an infeasible local minima, this is particularly true when including the tether model described in section 2.1.4 and when using the full model rather than the kinematic approximation.

To overcome this limitation we choose to implement a homotopy procedure, as described in [8]. The general idea behind that approach is to pick a less strongly non-linear optimization problem close to the one we want to solve and use the solution of that easier to solve problem as an initial guess for the full model. This is performed recursively as we gradually add in the strongly non-linear dynamics to a simplified model until we have a feasible solution for the strongly non-linear model. Essentially we start with "open" feedback loops and then close them gradually.

We choose to use as a near-linear model one where every force exerted on the plane is directly specified as a control input that we call $\boldsymbol{F}_{\text{fict}}$, this model can be described as follows:

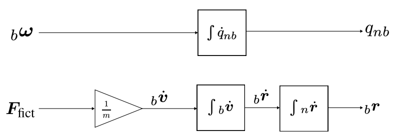

Fig. 9: Near-linear (kinematic) model block diagram. Notice that there is no feedback loop.

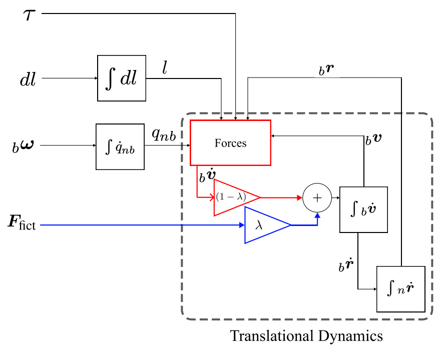

Fig. 10: Relaxed dynamical (kinematic) model block diagram. In red are denoted the subsystems that are ignored when $\lambda = 1$ and in blue the subsystems that are ignored when $\lambda = 0$

$$f_{\text{nl}}(\boldsymbol{x}_{\text{nl}}, \boldsymbol{u}_{\text{nl}}) = \begin{bmatrix} {}_b\dot{\boldsymbol{v}} \\ {}_n\dot{\boldsymbol{r}} \\ {}_{nb}\dot{q} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{m}\boldsymbol{F}_{\text{fict}} \\ q_{nb} \cdot {}_b\boldsymbol{v} \\ \frac{1}{2}q_{nb} \otimes q_\omega + \frac{1}{2}\lambda q_{nb}(\|q_{nb}\|^2 - 1) \end{bmatrix}. \tag{63}$$

This model has no feedback loop (essentially it is just integrating the controls over time) as is clearly visible in the block diagram figure 9.

In order to perform the homotopy procedure, we define a relaxed problem where a relaxation parameter $\lambda \in (0, 1)$ describes how much the dynamics are simplified (when $\lambda = 1$ the model is exactly the model described in eq. 63 and when $\lambda = 0$ the dynamics are the exact same as in eq. 49). The relaxed problem uses fictitious forces $\boldsymbol{F}_{\text{fict}}$ as system inputs additionally to the other inputs described in section 2.1.2, the influence of the forces given by the system dynamics from section 2.1 is linearly weighted by $(1 - \lambda)$ while the fictitious force's influence are linearly weighted by λ.

$$f_{\text{rel}}(\boldsymbol{x}_{\text{rel}}, \boldsymbol{u}_{\text{rel}}) = \begin{bmatrix} {}_b\dot{\boldsymbol{v}} \\ {}_n\dot{\boldsymbol{r}} \\ {}_{nb}\dot{q} \\ \dot{l} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{m}\left[(1-\lambda)(\sum {}_b\boldsymbol{F}_E - ({}_b\boldsymbol{\omega} \times {}_b\boldsymbol{v})) + \lambda \boldsymbol{F}_{\text{fict}}\right] \\ q_{nb} \cdot {}_b\boldsymbol{v} \\ \frac{1}{2}q_{nb} \otimes q_\omega + \frac{1}{2}\lambda q_{nb}(\|q_{nb}\|^2 - 1) \\ dl \end{bmatrix}. \tag{64}$$

Note that the tether dynamics $\dot{l} = dl$ are computed even when the model is fully relaxed, but that when $\lambda = 1$ they have no influence on the other variables, as is perhaps clearer on figure 10.

We define the relaxed problem $\mathcal{P}(\lambda)$ as the OCP described in eq. 51 using the relaxed dynamics defined in eq. 64 transcribed with collocation. We can then gradually lower the relaxation parameter γ, until it reaches 0, recursively using solutions of $\mathcal{P}(\lambda)$ as initial guesses to avoid unfeasible local minima following the algorithm 1.

2.2.6 Initial Guess Generation: Even with the homotopy procedure we found that finding an initial guess was non trivial, in the following section we present an approach to get a meaningful initial guess for the near-linear model from eq. 63. Our approach is the following : we solve a strongly constrained OCP where position and attitude is constrained for a large fraction of the colocation points, on both position and velocity. This gives an OCP where the solver only solves for inputs (fictitious forces and angular velocities) and has much less degree of freedom. This approach is able to generate state and input guesses that we then use to initialize the homotopy procedure.

```
Data: $\mathcal{P}$, $\Delta_{\text{rel}}$
Result: $(\mathcal{X}_{\text{opt}}, \mathcal{U}_{\text{opt}})$
generate an initial guess $(\mathcal{X}_{\text{init}}, \mathcal{U}_{\text{init}})$ (see section 2.2.6)
$\lambda \leftarrow 1$
$(\mathcal{X}, \mathcal{U}) \leftarrow (\mathcal{X}_{\text{init}}, \mathcal{U}_{\text{init}})$
while $\lambda > 0$ do
    initialize $P$ with $(\mathcal{X}, \mathcal{U})$
    $(\mathcal{X}, \mathcal{U}) \leftarrow$ solve $\mathcal{P}(\lambda)$
    $\lambda \leftarrow \lambda - \Delta_{\text{rel}}$
end
```

Algorithm 1: Homotopy Procedure

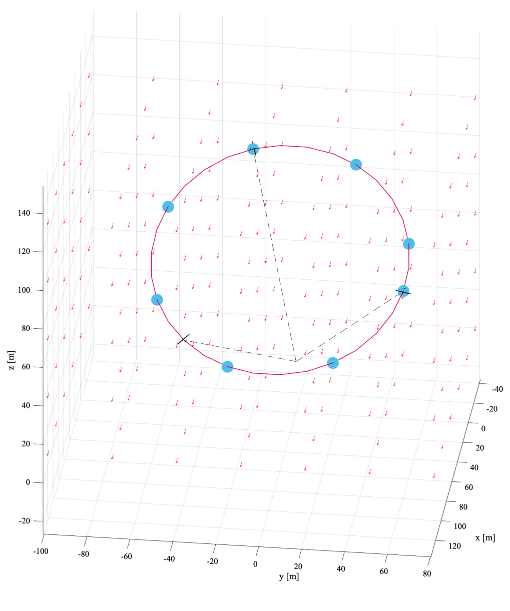

Fig. 11: Example of a strongly constrained OCP used to initialize the homotopy procedure. The red trajectory, is given as an initial guess and the speed and position is constrained as in equations 61a and 61b for all points denoted by blue dots.

3 Results

3.1 Model Validity

In the following section, we present several simulations generated using the full state model described in section 2.1.7. These simulations are generated using the *Matlab* implementation of the full model, which are integrated forwards with constant input using

Matlab's differential integration solver *ode45*. One of the main motivations for these experiments, is to determine whether or not the highly stiff nature of the differential equations which appears when the tether is introduced in the model leads to unphysical behavior in the system.

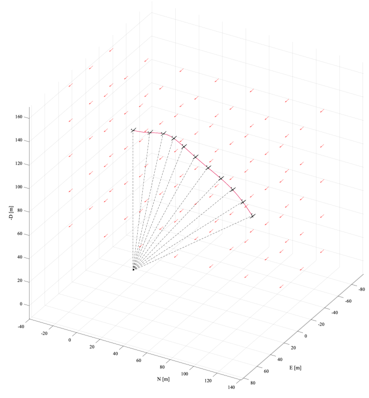

Fig. 12: Example evaluation trajectory where the kite flies into the tether (here with a $E = 9e9Pa$ Young moduli for the tether).

To evaluate that stiffness, we design a simulate a trajectory where the kite flies into the tether (see figure 12). We pick a high speed ($\approx 30m/s$) initial state (65) to ensure a high longitudinal force on the tether. The control vector is kept completely neutral during the entire trajectory, which we compute for $5s$.

$$\boldsymbol{x}_0 = \begin{bmatrix} {}_bv_x = 30.9748 \\ {}_bv_y = 0.0427 \\ {}_bv_z = -0.2747 \\ {}_b\omega_x = -0.0001; \\ {}_b\omega_y = 0.0001; \\ {}_b\omega_z = 0.0001; \\ {}_br_x = 0.1 \\ {}_br_y = 0.1 \\ {}_br_z = -120 \\ q_{bn,1} = 1 \\ q_{bn,2} = 0 \\ q_{bn,3} = 0 \\ q_{bn,4} = 0 \\ l = 120 \end{bmatrix}, \boldsymbol{u}_0 = \begin{bmatrix} \text{dE} = 0 \\ \text{dR} = 0 \\ \text{dA} = 0 \\ \tau = 0 \\ dl = 0 \end{bmatrix} \quad (65)$$

We then experiment with different tether stiffness values, and compare the simulated trajectories. To evaluate the quality of the solution we look at energy conservation in the system. Since aerodynamic forces are dissipative, and we do not account for loss, we would expect a good physical model to loose energy over time. This is precisely what happens with a low rigidity tether ($E = 9e9Pa$), when the rigidity is higher, artifacts appear when the tether force spikes (see figure 14 for a comparison of the system's energetic behavior at different tether stiffness values). Although the total energy of the system is not conserved during spikes for high Young modulus tethers, the artifacts that appear are transient and we do not see persistent energy being introduced in the system by the stiff behavior.

3.2 Homotopy Method

In the following section we present the sequence of OCP solutions that are recursively computed and used as initialization by the algorithm 1 (described in section 2.2.5) to get to a power-optimal solution. The main argument that algorithm 1 hinges on to improve convergence compared to a more naive approach is the following. Consider two instances $\mathcal{P}_1$ and $\mathcal{P}_2$ of the relaxed OCP (as defined in section 2.2) with two different relaxation parameters λ_1 and λ_2, what we hope for is that if $|\lambda_1 - \lambda_2|$ is sufficiently small, then $\mathcal{P}_1$ and $\mathcal{P}_2$ have feasible, locally optimal solutions $\mathcal{X}_1^*$ and $\mathcal{X}_2^*$ that are relatively close to each other. One way we can investigate wether or not this is the case is by looking at the solve times of the NLP solver (*IPOPT*) as it goes through algorithm 1 and solves relaxed subproblems $\mathcal{P}(\lambda)$. The evolution of the solver times is presented in figure 15.

Another metric we can use is the $L2$ norm of the distance between different iterations $\mathcal{X}_i, \mathcal{X}_j$. This metric has a more consistent behavior than the runtime metric and we observe that the difference between solutions converges to a small value as λ goes to 0. This is displayed in figure 16.

Finally, one can study the behavior of the algorithm by looking at the evolution of the geometry of the trajectories taken by the kite. Figure 17 shows the evolution of that geometry. In this case we observe that the trajectory converges to a stable value about two third of the way through the relaxation and then keeps a stable trajectory as it finishes to converge.

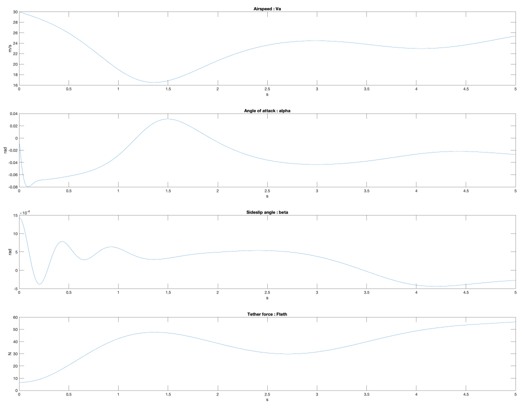

Fig. 13: Example state evolution for a trajectory where the kite flies into the tether (here with a $E = 9e9Pa$ Young Modulus in the tether, as in figure 12). Observe that the low speed is associated with a high angle of attack and that the tether force is not introduced until the kite hits the tether.

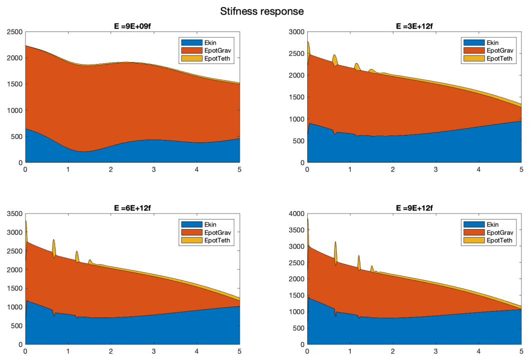

Fig. 14: Total energy plot of the system for 4 tether stiffness values ($1.0e12 \cdot [0.0091, 3.0227, 6.0363, 9.0500]Pa$), we observe that artifacts appear when the tether becomes stiffer. The artifacts are transient and we therefore do not consider them to invalidate the simulation results.

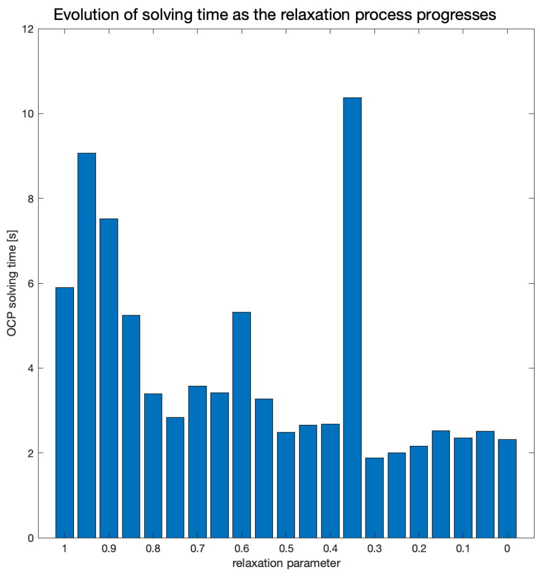

Fig. 15: Evolution of solve times for each OCP as the solver goes through iterations of algorithm 1. The x axis denotes the relaxation parameter λ. The algorithm goes from left to right as it lowers λ and one can observe that the general tendency is for the solve times to be become lower as the solution approaches convergence. There are nonetheless outliers (here one can observe that the solution for $\lambda = 0.35$, initialized with the $\lambda = 0.4$ solution, took an unexepectedly long time to be reached).

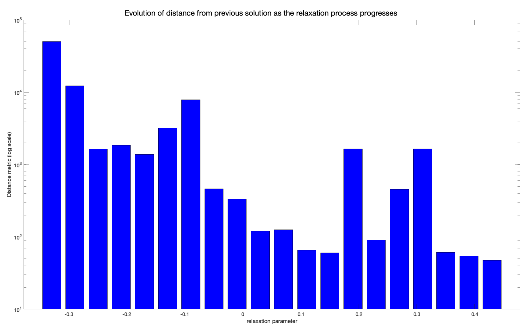

Fig. 16: Evolution of the distance (measured as L2 norms) between intermediate solutions as algorithm 1 progresses. The x axis denotes the relaxation parameter λ. The algorithm goes from left to right as it decreases λ and the y axis represent the distance between solutions.

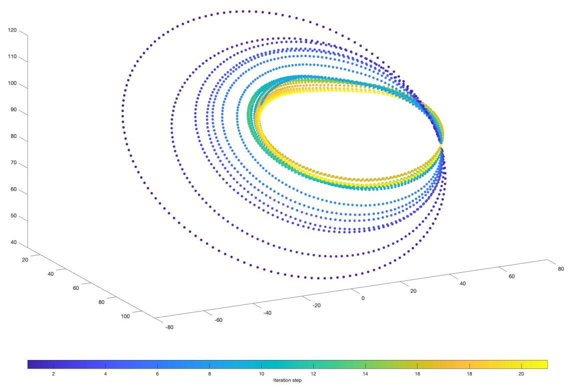

Fig. 17: Sequence of trajectories (position ${}_n\boldsymbol{r}$ over time)), as algorithm 1 progresses. Notice that one point is shared for all trajectories, this point is imposed as a path constraint (see section 3.3).

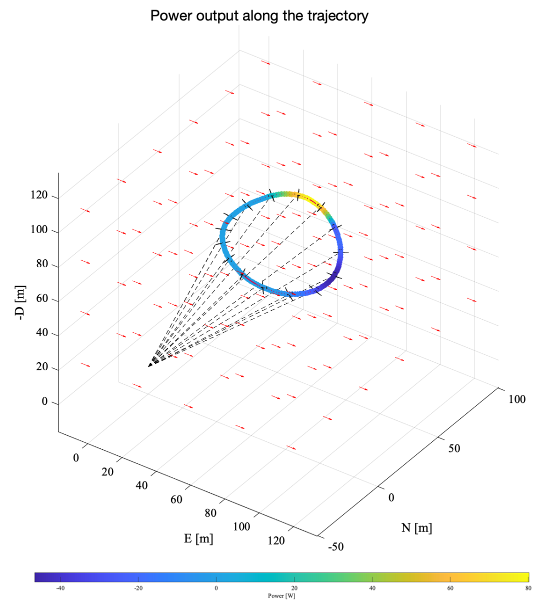

Fig. 18: Power optimized trajectory generated using the method presented in section 3.2. The trajectory's power output is overlaid on top of the kite's path and is bounded between $+80W$ and $-46W$ values.

3.3 Power Optimal Trajectories

In the following section we discuss a power optimal trajectory (represented in figure 18) generated using the OCP described in section 2.2 and solved with the homotopy procedure described in section 2.2.5. The wind vector has value ${}_n\boldsymbol{v}_W = [10, 0, 0]$ and corresponds to a $10m/s$ northward wind. The trajectory is power positive and generates a average power output $\hat{P} = 1.5258W$. It is computed over a $T = 15s$ period and the kite flies with an average $13.0017m/s$ ground speed (see system's state evolution figure 20). We use direct collocation with polynomials of order 2 and 10 segments to transcribe the OCP into a NLP.

The power generation periodically alternates between power-positive reel-out phases and power-negative reel-in phases. The mechanical power output P_{mec}, the longitudinal tether force $\|\boldsymbol{F}_{\text{lon}}\|$, the tether length l and the reel-in/out speed dl are shown in figure 19.

Although the optimization routine has the possibility to use thrust in it's solution, the thrust stays near 0 ($< 1\%$ of the maximum thrust) during the entire trajectory as can clearly be observed in figure 21.

A single position path constraint (as in equation 61a) is imposed on the solution. We constrain the optimization routine to pick a path starting at point ${}_n\boldsymbol{r}_0 = [58.5889, 67.5681, -74.7657]^T$, which is the first point in our rough initial guess trajectory. There is no other path constraint (no other position constraint, no orientation constraint, no velocity constraint etc.). That single path constraint is the minimum number of constraints required to get convergence, ideally it would be best to get the solver to work without that single constraint, but we did not manage to find a set of tuning parameters that allow us to drop that constraint. The following numerical parameters are used to generate the trajectory: $C_{\text{path}} = 1e1$, $C_{\text{control}} = 4e4$, $C_{\text{phys}} = 1e-3$ and $C_{\text{power}} = 1e3$. The R matrix for the linear

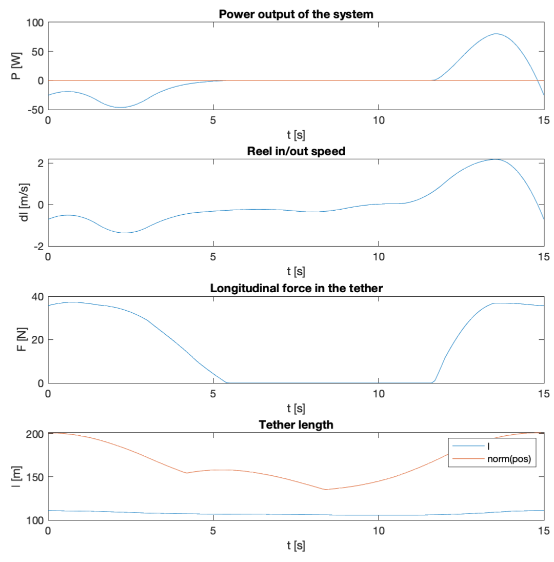

Fig. 19: Mechanical power output P_{mec}, longitudinal tether force $\|\boldsymbol{F}_{\mathrm{lon}}\|$, tether length l and reel-in/out speed dl. We can observe a $\approx 5s$ reel-in phase starting at the beginning of the period and a $4s$ reel-out phase starting around $12s$ into the period.

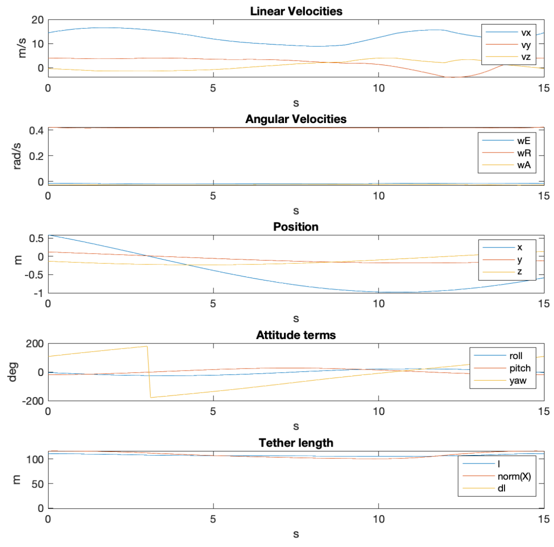

Fig. 20: State vector $\mathcal{X}^*$ of the AWE optimal trajectory plotted over time.

quadratic cost on control inputs is given as follows:

$$R = \begin{bmatrix} 6e3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6e3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6e3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1e3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1e1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{66}$$

The maximum tether elongation used in equation 61c is given by $\epsilon_{\max} = 0.05$. And the step value used for the homotopy procedure 1 is $\Delta_{\mathrm{rel}} = 0.05$. The following state and input constraints are imposed:

$$\boldsymbol{x}_{\mathrm{UB}} = \begin{bmatrix} \infty \\ 4 \\ 4 \\ \infty \\ \infty \\ \infty \\ \infty \\ \infty \\ \infty \\ \infty \\ 300 \end{bmatrix}, \qquad \boldsymbol{x}_{\mathrm{LB}} = \begin{bmatrix} 6 \\ -4 \\ -4 \\ -\infty \\ -\infty \\ -\infty \\ -\infty \\ -\infty \\ -\infty \\ -\infty \\ 0 \end{bmatrix}, \tag{67}$$

$$\boldsymbol{u}_{\mathrm{UB}} = \begin{bmatrix} 1.3963 \\ 0.7 \\ 1.3963 \\ 5 \\ 20 \\ \infty \\ \infty \\ \infty \end{bmatrix}, \qquad \boldsymbol{u}_{\mathrm{LB}} = \begin{bmatrix} -1.3963 \\ -0.7 \\ -1.3963 \\ 0 \\ -20 \\ -\infty \\ -\infty \\ -\infty \end{bmatrix}. \tag{68}$$

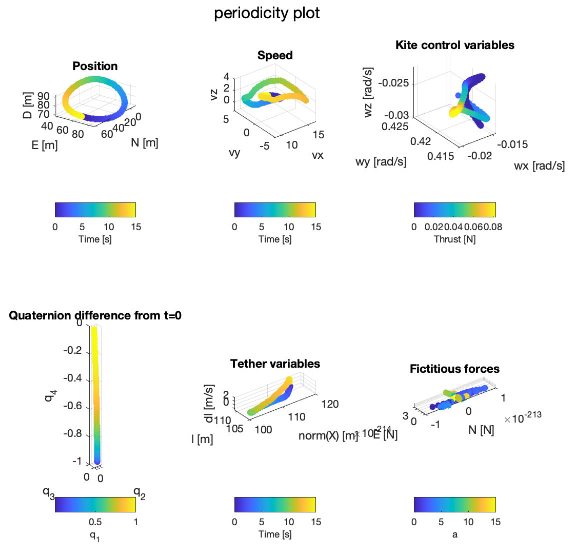

Fig. 21: Complete plot of system states $\mathcal{X}$ and inputs $\mathcal{U}$ as a 6 *3D* colored scatter plot collection.

4 Conclusion

In this project, we developed a method for generating power-optimal trajectories for a rigid-wing, lift-mode AWE system. This method successfully generates such trajectories, but is subject to several limitations that we detail in section 4.1 and that we think could be improved by investigating different aspects, which we list in section 4.2.

4.1 Limitations

The methods presented in this report does succeed in generating power-positive trajectories but the power output that we get is quite low ($1.5W$ in the trajectory generated in section 3.3). This can be attributed to several factors.

1. In order to get the trajectory to converge with our method, we are forced to impose a constraint on the position of one point along the trajectory (as discussed in section 3.3). This restricts the set of feasible power-positive trajectories that can be picked by the optimization routine.
2. Most lift-mode AWE concepts are designed to fly so-called "pumping cycle trajectories", where the kite gradually reels-out the tether over a long power positive phase before flying back towards the winch. Flying such a trajectory requires simulating solving a control problem with a much longer horizon than the one we consider in this project (the example that we present in section 3.3 has a $15s$ period whereas a pumping cycle could have a $> 1min$ period).

Furthermore, the relaxed model (described in section 2.2.5) that we considered is a relaxation of the kinematic model, not the full model. There is no reason to believe that the relaxation procedure doesn't generalize to the full model (it is demonstrated in [8] that the homotopy procedure is suitable to rotational dynamics similar to the ones we consider here, but our tether model is slightly more complex and creates parasitic force moments, which are not considered in [8]).

All control problems defined in this report assume a fixed time horizon (i.e. the period of the trajectory is given by the user, and can be though of as an hyper-parameter). This is a limitation as the actual optimal trajectory period is unknown.

4.2 Further work

The most obvious improvement to be made to the results presented in this report is the implementation of the relaxation procedure for the full model instead of the kinematic model. As this is expected to make the transcribed NLPs even more strongly non linear and non-convex than they are in the current implementation, investigating optimization techniques to enable more robust convergence seems like an interesting route to follow. One recurring feature that seemed to lead to non-convergence is oscillations between local minima, perhaps investigating the introduction of a regularization term would be relevant.

Furthermore, the reformulation of the OCP in such a way that the time horizon is an (explicit or implicit) decision variable would probably enable better quality solutions.

Finally, and perhaps more importantly, the implementation of a proper formulation for pumping cycle trajectories (perhaps as something approaching a multi-phase trajectory optimization problem), would make the techniques described in this report much more useful to a practical implementation of AWE. The current implementation can already be used to generate pumping cycle trajectories (see figure 22), but doing so requires manually constraining the initial and terminal states of the kite for each pumping cycle. This makes it especially hard to find power-optimal solutions, and we expect that an approach that allows to optimize over a full pumping cycle would yield much better results.

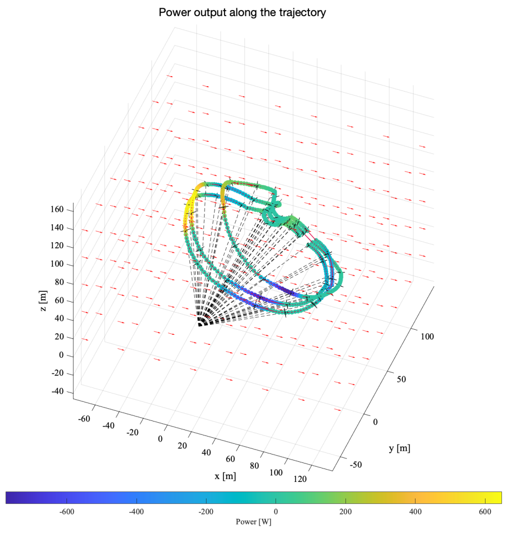

Fig. 22: Example pumping cycle solution generated with the current implementation. Here states final and terminal states are imposed as constraints in the OCP.

References

[1]Chunhui Chen and O. Mangasarian. "A Class of Smoothing Functions for Nonlinear and Mixed Complementarity Problems". In: *Computational Optimization and Applications* 5 (Mar. 1996), pp. 97–138. DOI: 10.1007/BF00249052.

[2]P.R. Amestoy et al. "A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling". In: *SIAM Journal on Matrix Analysis and Applications* 23.1 (2001), pp. 15–41.

[3]Andreas Wächter and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106 (2006), pp. 25–57.

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

[5]Basile Graf. *Quaternions and dynamics*. 2008. arXiv: 0811.2889 [math.DS].

[6]Ivan Argatov and Risto Silvennoinen. "Airborne Wind Energy". In: ed. by Uwe Ahrens, Moritz Diehl, and Roland Schmehl. 2013. Chap. Efficiency of traction power conversion based on crosswind motion, pp. 65–80. ISBN: 978-3-662-50879-4.

[7]Moritz Diehl. "Airborne Wind Energy". In: ed. by Uwe Ahrens, Moritz Diehl, and Roland Schmehl. 2013. Chap. Airborne Wind Energy: Basic Concepts and Physical Foundations, pp. 3–22. ISBN: 978-3-662-50879-4.

[8]Sébastien Gros, M. Zanon, and Moritz Diehl. "A relaxation strategy for the optimization of Airborne Wind Energy systems". In: *2013 European Control Conference (ECC)*. 2013, pp. 1011–1016. DOI: 10.23919/ECC.2013.6669670.

[9]Mario Zanon, Sébastien Gros, and Moritz Diehl. "Airborne Wind Energy". In: ed. by Uwe Ahrens, Moritz Diehl, and Roland Schmehl. 2013. Chap. Model Predictive COntrol of Rigid-Airfoil Airborne Wind Energy Systems, pp. 219–233. ISBN: 978-3-662-50879-4.

[10]Antonello Cherubini et al. "Airborne Wind Energy Systems: A review of the technologies". In: *Renewable and Sustainable Energy Reviews* 51 (2015), pp. 1461–1476. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2015.07.053. URL: https://www.sciencedirect.com/science/article/pii/S1364032115007005.

[11]Sebastien Gros, Marion Zanon, and Moritz Diehl. "Baumgarte stabilisation over the SO(3) rotation group for control". In: *2015 54th IEEE Conference on Decision and Control (CDC)*. 2015, pp. 620–625. DOI: 10.1109/CDC.2015.7402298.

[12]Brian Stevens. "Aircraft control and simulation : dynamics, controls design, and autonomous systems". In: Hoboken, New Jersey: Wiley, 2015, p. 27. ISBN: 9781118870990.

[13]Matthew Kelly. "An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation". In: *SIAM Review* 59 (Jan. 2017), pp. 849–904. DOI: 10.1137/16M1062569.

[14]Joel A E Andersson et al. “CasADi – A software framework for nonlinear optimization and optimal control”. In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36. DOI: 10.1007/s12532-018-0139-4.

[15]G. Licitra et al. “Performance assessment of a rigid wing Airborne Wind Energy pumping system”. In: *Energy* 173 (2019), pp. 569–585. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2019.02.064. URL: https://www.sciencedirect.com/science/article/pii/S0360544219302592.

[16]Johannes Waibel. “Flight Path Optimization and Predictive Control of an Airborne Wind Energy System”. MA thesis. University of Stuttgart, EPFL, 2020.

[17]*Ampyx Power*. 2022. URL: https://www.ampyxpower.com/.

[18]*Kitemill*. 2022. URL: https://www.kitemill.com/.

[19]*TwingTec*. 2022. URL: https://twingtec.ch/.