Optimal Control for Multi-Kite Emergency Trajectories

Thilo Bronnenmeyer

Masterarbeit

2018

Universität Stuttgart
Institut für Flugmechanik und Flugregelung

Kiteswarms GmbH
Am Flugplatz 1, 79427 Eschbach

Erklärung

Hiermit versichere ich, dass ich diese Masterarbeit selbständig mit Unterstützung der Betreuer angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Die Arbeit oder wesentliche Bestandteile davon sind weder an dieser noch an einer anderen Bildungseinrichtung bereits zur Erlangung eines Abschlusses eingereicht worden. Die Arbeit wurde weder vollständig noch in Teilen bereits veröffentlicht. Das elektronische Exemplar stimmt mit den anderen Exemplaren überein.

Ich erkläre weiterhin, bei der Erstellung der Arbeit die einschlägigen Bestimmungen zum Urheberschutz fremder Beiträge entsprechend den Regeln guter wissenschaftlicher Praxis[1] eingehalten zu haben. Soweit meine Arbeit fremde Beiträge (z.B. in der Form von Bilder, Zeichnungen, Textpassagen etc.) enthält, habe ich diese Beiträge als solche gekennzeichnet (Zitat, Quellenangabe) und eventuell erforderlich gewordene Zustimmungen der Urheber zur Nutzung dieser Beiträge in meiner Arbeit eingeholt. Mir ist bekannt, dass ich im Fall einer schuldhaften Verletzung dieser Pflichten die daraus entstehenden Konsequenzen zu tragen habe.

Ort, Datum	Thilo Bronnenmeyer

Betreuer: Prof. Dr.-Ing. Walter Fichter, Dr. rer. nat. Werner Grimm
Die Arbeit entstand bei:
Kiteswarms GmbH,
Am Flugplatz 1,
79427 Eschbach

[1] Nachzulesen in den DFG-Empfehlungen zur „Sicherung guter wissenschaftlicher Praxis“ bzw. in der Satzung der Universität Stuttgart zur “Sicherung der Integrität wissenschaftlicher Praxis und zum Umgang mit Fehlverhalten in der Wissenschaft“

Kurzfassung

Die vorliegende Masterarbeit beschäftigt sich mit Optimalsteuerung für Multi-Kite-Notfalltrajektorien. Multi-Kite-Systeme bestehen aus kleinen autonomen Flugzeugen (üblicherweise Kites genannt), die durch Seile untereinander und mit einer Bodenstation verbunden sind. Sie werden im Bereich der Höhenwindenergie zur Energieerzeugung verwendet. Das Ziel dieser Masterarbeit ist es, eine modulare Struktur für Optimalsteuerungsprobleme zu erstellen, die benutzt werden kann, um optimale Notlandungstrajektorien für Multi-Kite-Systeme zu berechnen. Darin inbegriffen ist die systematische Kategorisierung möglicher Notfallszenarien und die Entwicklung von Strategien, mit denen ihnen begegnet werden kann. Wir formalisieren eine modulare Homotopiestrategie, um eine gute und zulässige Initialisierung für die komplexen Optimalsteuerungsprobleme zu finden. Des Weiteren formulieren wir verschiedene Optimalsteuerungsprobleme, die jeweils unterschiedliche Notfalltszenarien repräsentieren. Wir analysieren ihre jeweiligen Lösungen sowohl in Bezug auf die physikalischen Phänomene, die dafür sorgen, dass diese bestimmten Trajektorien optimal sind, als auch in Bezug auf die Sensitivität der optimalen Trajektorien hinsichtlich einer Änderung von physikalischen und numerischen Parametern. Die Optimalsteuerungsprobleme sind implementiert in die `python`-Toolbox `AWEbox`.

Abstract

This thesis concerns itself with optimal control for multi-kite emergency trajectories. Multi-kite systems consist of small autonomous airplanes (usually referred to as kites) that are linked to each other and to the ground by tethers and are employed in the field of airborne wind energy to generate power. The goal of this thesis is to construct a modular optimal control framework that can be used to compute optimal emergency landing trajectories for multi-kite systems. This framework includes a systematical categorization of possible emergency scenarios and the development of strategies in order to deal with each of those categories. We formalize a modular homotopy strategy to find good and feasible initial guesses for the complex nonlinear optimal control problems. We further formulate a number of different optimal control problems, each representing different emergency scenarios, and analyze their corresponding solutions, both in terms of what physical phenomena make these particular trajectories optimal and how the optimal trajectories change when varying physical or numerical parameters. The optimal control framework is implemented within the `python` toolbox `AWEbox`.

Thanks and Acknowledgements

I would like to thank all my supervisors at the University Stuttgart, the University of Freiburg as well as Kiteswarms for their help and encouragement. Thanks to Prof. Walter Fichter, Dr. Walter Grimm, Prof. Moritz Diehl and Reinhart Paelinck for making this master thesis possible. Special thanks to my fellow `AWEbox`-developers Rachel Leuthold and Jochem De Schutter as well as to the whole Kiteswarms team for the great work together, both in the past and the days to come!

Contents

Erklärung iii

Kurzfassung / Abstract v

Thanks and Acknowledgements vii

List of Figures xi

List of Tables xiii

Nomenclature xv

1. Introduction 1
- 1.1. Multi-Kite Airborne Wind Energy Systems . . . 3
- 1.2. Dual-Kite Pumping Cycles . . . 4
- 1.3. Emergency Scenarios for Airborne Wind Energy Systems . . . 7
- 1.4. Related Work . . . 8
- 1.5. Organization of Contents . . . 9

2. Optimal Control Methods and Theory 11
- 2.1. System Modeling with DAEs . . . 11
- 2.2. Direct Collocation . . . 13
- 2.3. Interior-Point Methods for Numerical Optimization . . . 18
- 2.4. Homotopy Strategy . . . 22

3. Emergency Response Concept 29
- 3.1. Expected Emergency Scenarios . . . 29
- 3.2. Emergency Detection . . . 30
- 3.3. Hierarchical Emergency Landing Strategies . . . 33

4. DAE System Model 35
- 4.1. System States, Controls and Coordinate System . . . 35
- 4.2. Lagrangian Mechanics . . . 37
- 4.3. Wind and Atmospheric Model . . . 38
- 4.4. Generalized Forces Acting on the System . . . 39
  - 4.4.1. Aerodynamic Kite Forces . . . 39

4.4.2. Tether Drag . . . 40
4.5. On-board Battery Model . . . 43

5. Optimal Control Problem Formulation **47**
5.1. Cost Function . . . 47
5.2. Constraints . . . 49
5.2.1. Equality Constraints . . . 49
5.2.2. Inequality Constraints . . . 51
5.3. Homotopy Schedule . . . 55
5.4. Initial Guess Generation . . . 58
5.4.1. Initial and Terminal Points . . . 58
5.4.2. Consistent Interpolation . . . 59
5.5. Numerical Issues . . . 61
5.5.1. Infeasibility of Parameterized Trajectories . . . 61
5.5.2. Acceleration Regularization . . . 61
5.5.3. Homotopy Meta-Parameter Tuning . . . 62
5.5.4. Stagger Distance . . . 64

6. Software Implementation **67**
6.1. `AWEbox` . . . 67
6.2. `CasADi` and `IPOPT` . . . 69

7. Landing Trajectories with Nominal Flight Behavior **71**
7.1. OCP Formulation and Homotopy Schedule . . . 71
7.2. Analysis of Solutions . . . 73
7.2.1. Homotopy Steps . . . 73
7.2.2. Sensitivity to Wind Speed . . . 81
7.2.3. Potential and Kinetic Energy . . . 84

8. Overview of Compromised Landing and Transition Trajectories **87**
8.1. Transition Trajectories . . . 87
8.2. Emergency Landings with Compromised Flight Behavior . . . 89
8.2.1. Actuator Malfunction . . . 89
8.2.2. On-Board Battery Malfunction . . . 92
8.2.3. Structural Damage . . . 93

9. Conclusions and Outlook **99**

A. Appendix A **103**

B. Appendix B **107**

Bibliography **109**

List of Figures

1.1. Transformation of a towered wind turbine into an AWES [9]. 2
1.2. Classification of AWESs according to [9]. Red star indicating the system configuration this thesis focuses on. 4
1.3. Schematic sketch of a dual-kite AWES [9]. 5
1.4. Power generating pumping cycle trajectory of a dual-kite AWES. 6

3.1. State machine for error detection. 32

4.1. Schematic sketch of the dual-kite AWES including the coordinate system and the labeling of the system's nodes [24]. 36
4.2. Schematic sketch of the aerodynamic forces $\mathbf{F}_\mathrm{L}$ and $\mathbf{F}_\mathrm{D}$. 40

5.1. Infeasibility of parameterized trajectories. Left: Comparison of spline approximation (blue) and actual curve (red) for C_L. Right: Zoomed-in detail of overshoot phenomenon with dashed line indicating the bound on C_L. . 62
5.2. Acceleration Regularization. Comparison of the number of iterations, NLP construction time, optimization time and maximum invariants for different values of $\hat{R}_{\ddot{\mathbf{q}}}$. 63
5.3. Homotopy meta-parameter tuning. Comparison of solutions after nominal landing homotopy step for $\tau_\mathrm{b} = 1 \cdot 10^{-2}$ (red) and $1 \cdot 10^{-5}$ (blue). Initial trajectory displayed in black. Position of the ground-station represented by a red dot. 64

6.1. Schematic overview of the `AWEbox` on `sweep` and `trial` level. 70

7.1. Nominal landing trajectory. Solution after initial homotopy step compared to initial guess (dashed). The initial trajectory is displayed in black. Position of the ground-station represented by a red dot. 74
7.2. Nominal landing trajectory. Solution after tracking homotopy step compared to solution of previous homotopy step (dashed). Initial trajectory displayed in black. Position of the ground-station represented by a red dot. 75
7.3. Nominal landing trajectory. Solution after nominal landing homotopy step compared to solution of previous homotopy step (dashed). Initial trajectory displayed in black. Position of the ground-station represented by a red dot. 76

7.4. Nominal landing trajectory. Comparison of trivial (red), projected (blue) and equivalence (green) tether drag model. Initial trajectory displayed in black. Position of the ground-station represented by a red dot. 77
7.5. Nominal landing trajectory. Comparison of the accelerations $a_{\mathcal{A}}$, a_1 and $a_{\mathcal{N}}$ for the trivial, projected and equivalence tether drag models. 79
7.6. Nominal landing trajectory. Comparison of the number of iterations, NLP construction time, optimization time and maximum invariant for the trivial, projected and equivalence tether drag models. 80
7.7. Nominal landing trajectory. Solution after final homotopy step with equivalence tether drag model compared to solution of previous homotopy step (dashed). Initial trajectory displayed in black. Position of the ground-station represented by a red dot. 82
7.8. Nominal landing trajectory. Sensitivity to change in u_{ref}. Comparison of average terminal node velocity, maximum and average tether tension and optimal initial trajectory parameter. 83
7.9. Nominal landing trajectory. Kinetic and potential energy over time for individual nodes and whole system. 84

8.1. Transition trajectory. Initial and terminal trajectories displayed in black. Position of the ground-station represented by a red dot. 89
8.2. Landing trajectory with compromised actuation. Finding feasible trajectories when varying f_{c} and ξ_0. 91
8.3. Landing trajectory with structural damage. Illustration of structural damage modeling with assumed constraint set $\mathcal{D}$, real constraint set $\mathcal{W}$ and nominal constraint set $\mathcal{P}$. 96
8.4. Landing trajectory with structural damage. Sensitivity to change in f_{c}. Comparison of maximum tether tension, optimization time, average terminal node velocity and maximum invariants. 97

List of Tables

2.1. Generic homotopy schedule. 25
2.2. Barrier parameters for $\mathcal{P}_l$. 26

4.1. Model states. 37

7.1. Nominal landing homotopy schedule. 73

8.1. Transition homotopy schedule. 88
8.2. Actuator malfunction homotopy schedule. 91
8.3. On-board battery malfunction homotopy schedule. 93
8.4. Structural damage homotopy schedule. 95

A.1. Model parameters. 103
A.2. Model bounds. 104
A.3. Numerical parameters. 105

Nomenclature

General Notes on the Notation

Derivatives with respect to time are denoted with dots above the variable. Partial derivatives with respect to a variable a are denoted by $(\cdot)_a$. Important indices are designated by unique letters (cf. the list of indices in this section). All other indices are denoted as $(\cdot)_{i_{(\cdot)}}$. Vectors and matrices are presented in bold font to distinguish them from scalar expressions. Collocated variables are denoted $(\cdot)_{i,j}$.

Abbreviations

AWE	Airborne Wind Energy
AWES	Airborne Wind Energy System
AR	Artificial Reasoning
DOF	Degree Of Freedom
DAE	Differential Algebraic Equation
DE	Differential Equation
DP	Dynamic Programming
IP	Interior-Point
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
LSE	Linear System of Equations
NLP	Nonlinear Program
OC	Optimal Control
OCP	Optimal Control Problem
QP	Quadratic Program
SQP	Sequential Quadratic Programming
UAV	Unmanned Aerial Vehicle

Important Variables

$\mathbf{x}$	state vector
$\mathbf{u}$	control vector
E	generated energy
$\boldsymbol{\lambda}$	algebraic variable vector
t	time
$\mathbf{p}$	homotopy parameter vector
J	cost function
$\mathbf{q}$	node position/generalized coordinate vector
C_L	lift coefficient
Ψ	roll angle
l_T	main tether length
$\mathbf{F}_\mathrm{A}$	aerodynamic forces vector
$\mathbf{F}_\mathrm{L}$	lift force vector
$\mathbf{F}_\mathrm{D}$	drag force vector
$\mathbf{h}$	inequality constraint vector
$\mathbf{c}$	equality constraint vector
$\mathbf{g}$	algebraic equations

Important Indices

$i \in \mathcal{I}$	index of collocation interval
$j \in \mathcal{J}$	index of control points
$p \in \mathcal{N}$	index of system nodes
$a \in \mathcal{A}$	index of kite nodes
$k \in \mathcal{T}$	index of tethers
$l \in \mathcal{P}_\mathrm{H}$	index of homotopy steps

1. Introduction

This thesis will concern itself with optimal control for multi-kite emergency trajectories. Optimal control is meant in the sense of offline open-loop trajectory optimization, while the term multi-kite describes a system consisting of multiple small aircraft that are tethered together and are used in the field of airborne wind energy (AWE). The following introduction shall serve to clear up these terms.

For conventional towered wind turbines, the outer 30 % of the blades generate more than half of the total power [9]. The outer tips of the blades however only make up a very small percentage of the overall mass and production cost of a towered wind turbine. Ironically, the scale of towered wind turbines is not restricted by the rotors, but by the moment generated at the bottom of the tower. Thus, the tower is not only inefficient and expensive, but also the bottleneck of the whole structure when trying to scale it up. Hence, the following question arises: Is it possible to build a wind turbine that only consists of the efficient parts of the blades, while omitting the large and expensive towers?

A possible answer to this question is the concept of AWE. Airborne wind energy systems (AWES) generate power by substituting autonomous aircraft of various kinds for the rotors of the conventional wind turbine and a tether for the tower itself. Such aircraft include planes and soft-kites, which are usually tethered to the ground in order to use the relative velocity of the ground and the air to generate energy. This concept is visualized in Fig. 1.1. Both planes and soft-kites are usually referred to as kites in an AWE context. In terms of the forces they can compensate versus the amount of material and energy that is needed to produce them, tethers are much more efficient than concrete structures. Also, a tether is much more flexible than concrete, which is why the AWESs can be designed in a way that only tether stress but no moment is generated. This is a huge advantage, since it gets rid of the structural bottleneck that is the base and the foundation of the tower that must be able to withstand large moments over an extended period of time. Overcoming this bottleneck enables AWESs to reach higher altitudes, where the winds are usually stronger and more consistent than closer to the ground [4]. Thus, AWE has the potential to be both cheaper and more efficient than conventional towered wind turbines. In order to quantify this, we introduce the power harvesting factor

$$\zeta = \frac{P_\mathrm{h}}{P_\mathrm{w}}, \tag{1.1}$$

which describes the fraction of the power P_h that can be harvested with a given wing surface and the wind power P_w that flows through an area of the same size. The ζ-factor of towered wind turbines is approximately 5.5, while ζ-factors of up to 8 have been

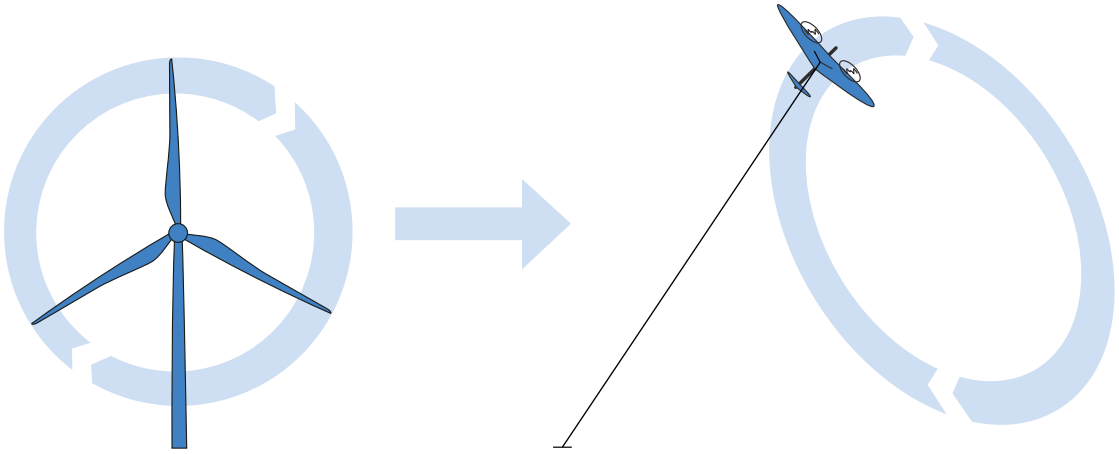

Figure 1.1.: Transformation of a towered wind turbine into an AWES [9].

measured for AWESs [9].

The concept of AWE has been around since the 1970s, most notably with the American engineer Miles Loyd, who first investigated the concept of power generation by tethered flight in detail [28]. He was also the first to come up with an estimation of the maximum steady state power P that can theoretically be generated by an AWES, namely

$$P = \frac{2}{27}\rho A v_{\mathrm{w}}^3 C_{\mathrm{L}} \left(\frac{C_{\mathrm{L}}}{C_{\mathrm{D}}}\right)^2, \tag{1.2}$$

with wing area A, wind speed v_{w}, lift and drag coefficients C_{L} and C_{D}, and air density ρ. Although this is only an approximation under idealized assumptions, it sparked interest concerning the large quantities of power that can potentially be generated with AWESs given the right system configuration. This is due to the concept of crosswind kite power: A tethered kite that is flown fast in crosswinds results in a large increase in the resulting tether tension, a phenomenon widely used by e.g. kite-surfers to lift themselves up into the air. Note that (1.2) depends on the third power of the wind speed v_{w}, which emphasizes the huge potential of crosswind power.

Now that the basic concepts of AWE have been elaborated, section 1.1 will go into details about multi-kite AWESs specifically. Section 1.2 goes on to explain the shape and properties of dual-kite pumping cycles. Section 1.3 then gives an introduction to the possible emergency scenarios regarding multi-kite AWESs, while section 1.4 gives an overview over the research that has already been done in relation to the topic of this thesis. Lastly, section 1.5 lays out how the thesis' content will be structured.

1.1. Multi-Kite Airborne Wind Energy Systems

There are several ways to classify AWESs. A Venn diagram of the classification done in [9] is shown in Fig. 1.2[1]. One distinction is between systems with on-board power generation and those with ground-based power generation. The first class generates power by carrying a generator with a turbine as part of the kite. By flying so called power cycles, usually periodic figure of eight or circular shapes, the AWES drives the on-board generators and thus produces power. The second class uses the relative velocity of the ground and the kite to generate power. To do so, the kite's tether is connected to a generator and periodically reeled in and out. If the kite's trajectory is planned smartly, the reel-out phase generates much more power than the reel-in phase consumes. The power cycle of a ground-based AWES is usually referred to as a pumping cycle. The setup that is investigated in this thesis uses ground-based power generation. It is obvious that the efficiency of an AWES, both with on-board and ground-based power generation, is highly dependent on the choice of the power cycle. This is even more true for an AWES with ground-based power generation, since the system needs to convert the air's relative velocity to the ground into tension of the tether that is connected to the generator. Also, the system has an extra degree of freedom (DOF) regarding the reel-in profile of the main tether.

The next distinction is that between rigid and flexible structures. Flexible structures such as soft-kites can be built extremely light-weight and do not cause damage when crashing into the ground. Rigid structures generally exhibit higher lift to drag ratios and can thus generate more power. The combination of flexible structures together with on-board power generation is not useful, since the weight of an on-board generator makes the main advantage of flexible structures, their extremely light weight, obsolete.

Another distinction is the number of kites connected to one main tether and thus to one ground-station. Single kite systems need a long main tether to reach high altitude winds. A long tether however also produces large tether drag and vibrations as it is moving around. One possibility to get around this problem is to connect multiple kites with relatively short secondary tethers to a single longer main tether that is then in turn connected to the ground-station. The length of the main tether enables the AWES to reach high altitude winds, while the shorter secondary tethers allow the kites to perform fast maneuvers without causing a lot of tether drag [45]. It is intuitive to choose two kites for the multi-kite system since it constituted the lowest degree of complexity for such a system. This kind of multi-kite AWES is typically called a dual-kite AWES. A sketch of the dual-kite system is shown in Fig. 1.3. Even though most research interest regarding multi-kite systems has focused on these dual-kite AWESs, triple kites or more complex configurations might have advantageous structural properties, which have so far been left undiscovered.

This thesis will focus on dual-kite systems with ground-based power generation as

[1]For a more detailed overview of different AWESs, we refer to [8].

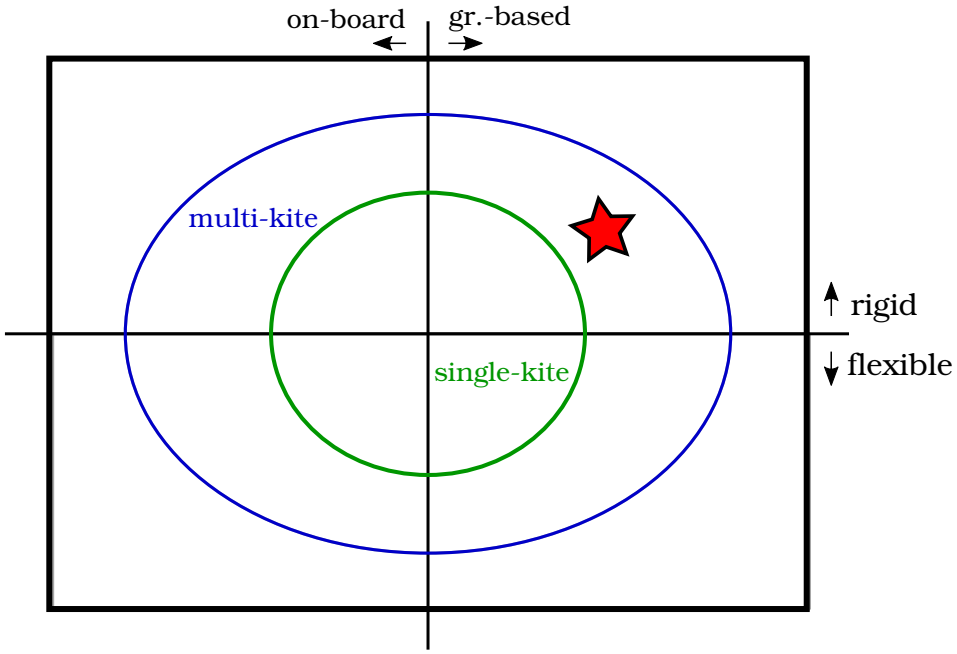

Figure 1.2.: Classification of AWESs according to [9]. Red star indicating the system configuration this thesis focuses on.

indicated by the red star in Fig. 1.2. The controls of such a system usually comprise a combination of actuation via the tether, i.e. by using the generator to reel the main tether in or out, and flaps as well as rotors on the kites. Dual kites already exhibit a high degree of complexity and have not been researched in an exhaustive manner, especially considering any topic not directly related to the computation of power cycles. The jump in complexity of the problem from single to dual kites is very large, since the number of states required to describe the model almost doubles and the AWES is able to perform much more sophisticated maneuvers, as already explained above. With increasing complexity of the multi-kite AWES's configuration, the analytical and numerical effort in researching the system also increases sharply. Hence, it is a prudent approach to first develop solutions for the less computationally demanding dual-kite AWESs before advancing to triple-kite AWESs or more complex configurations. Thus, all results in this thesis will be formulated in a way that is easy to generalize to different multi-kite configurations.

1.2. Dual-Kite Pumping Cycles

In section 1.1 we briefly touched on the topic of pumping cycles. A pumping cycle is the periodic orbit of an AWES with ground-based power generation. Since the multi-kite

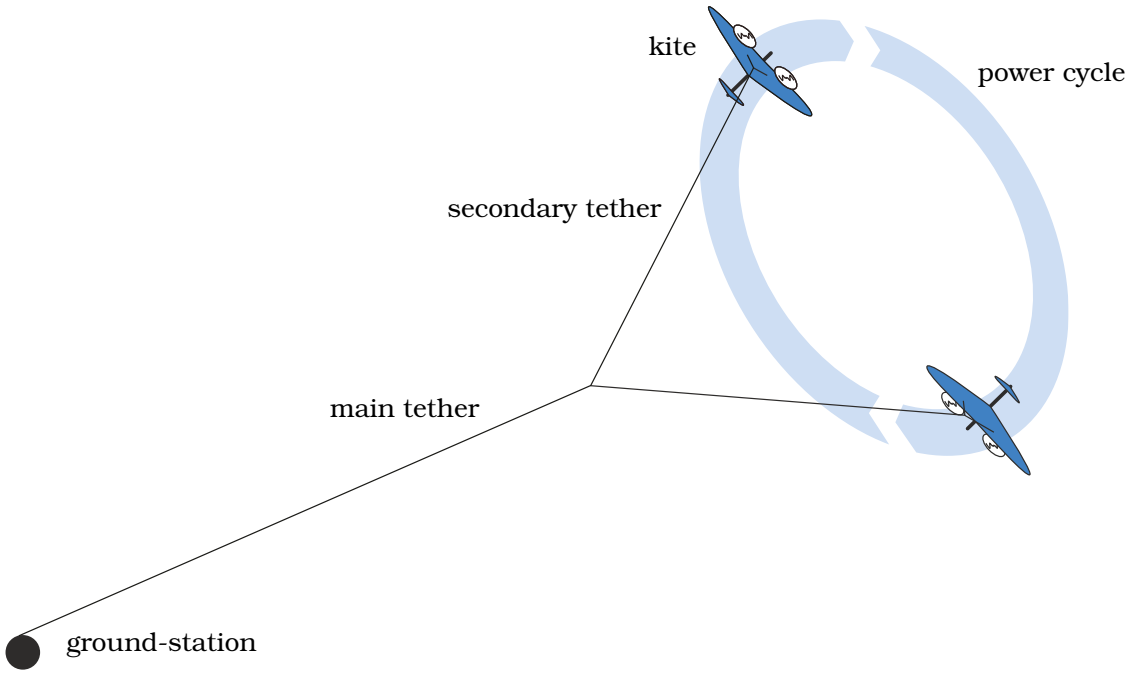

Figure 1.3.: Schematic sketch of a dual-kite AWES [9].

emergency trajectories discussed in this thesis all start on pumping cycles, it is necessary to give more details about their shape and properties. In the literature, pumping cycles are usually derived by solving an optimal control problem (OCP) (cf. [24] for dual-kite and [19] for single-kite AWESs). The OCP formulation usually involves a periodicity condition in the constraints and a term in the cost function that represents the amount of generated power normalized with the period length. By following [24], we can compute the optimal pumping cycle shown in Fig. 1.4[2]. The plot shows four different projections of the same trajectory, including an isometric view. The red and green lines represent the positions of the two kites. Only the trajectory of the two kite nodes is displayed in order to keep the plot clear and easy to read. In the isometric projection, the implicit orientation of the kites is shown with little planes. All further trajectory plots in this thesis are presented in the same fashion.

The shape of the optimal pumping cycle shows the distinctive separation of reel-in and reel-out phase. During the reel-out phase, the kites fly on a looping trajectory away from the ground-station, thereby reeling out the main tether and generating power. The loopings are flown with a high lift coefficient in order to increase the tension on the main tether and thus increase the generated power.

During the reel-in phase, the kites fly outwards and back towards the ground-station.

[2]For details of the OCP formulation, we refer to [24].

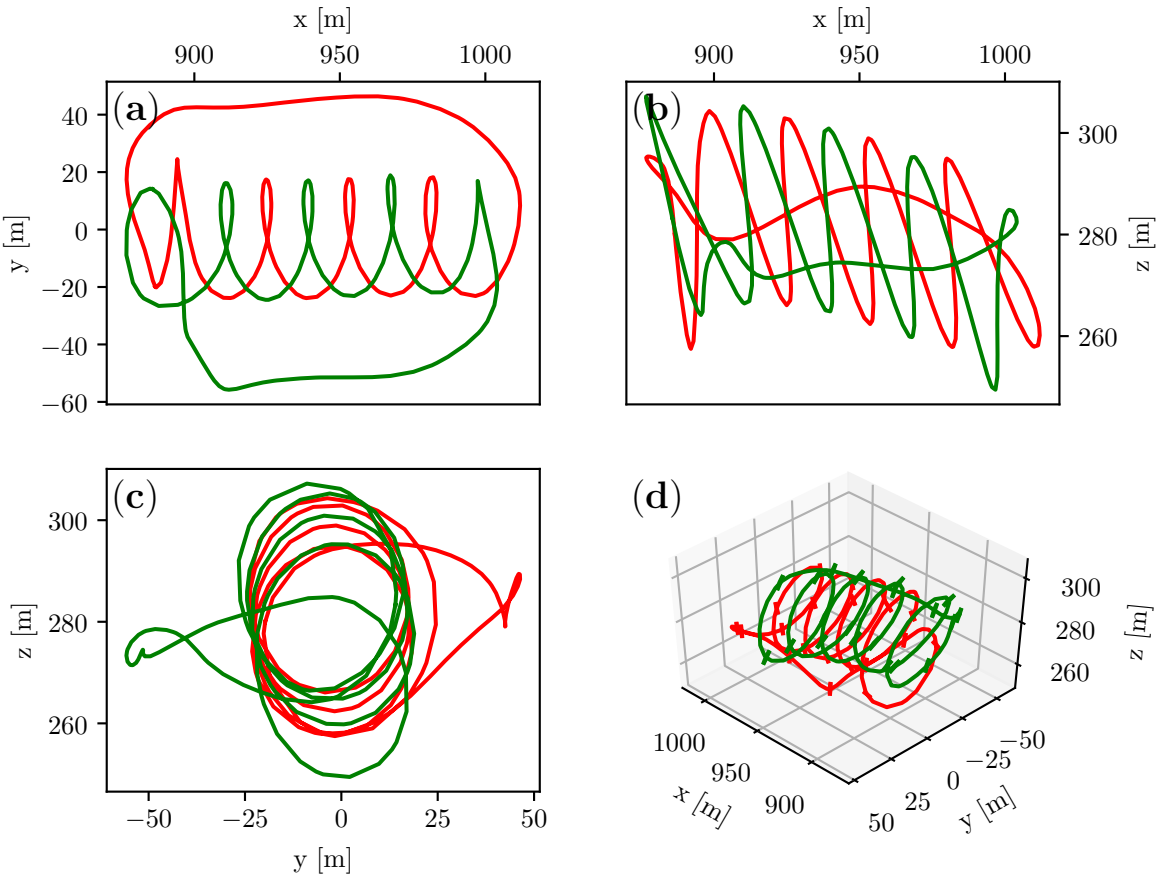

Figure 1.4.: Power generating pumping cycle trajectory of a dual-kite AWES.

Flying outwards decreases the projection of the kites' aerodynamic forces on the main tether, and thus also the amount of power that is needed to reel the kites in. For the same reason, the kites' lift coefficient is decreased during the reel-in.

The center node of the dual-kite AWES ideally only moves back and forth in the direction of the main tether in order to drive the generator on the ground. All crosswind maneuvers of the kites that are orthogonal to the direction of the main tether only affect the motion of the secondary tethers. As mentioned in section 1.1, this is one of the main incentives to introduce multi-kite configurations for AWESs.

In order to ensure that the optimal pumping trajectory only consists of a single reel-in and a single reel-out phase, a technique called phase-fixing is employed. This means that the overall time interval $T = [0, t_\mathrm{f}]$ is divided into two intervals $T_1 = [0, p_\mathrm{fix} t_\mathrm{f}]$ and $T_2 = [p_\mathrm{fix} t_\mathrm{f}, t_\mathrm{f}]$ with the phase fixing parameter $p_\mathrm{fix} \in [0, 1]$. Intervals T_1 and T_2 are designated for the reel-out and reel-in phase respectively. To ensure that the phases actually happen during their designated time intervals, a constraint is added such that

the tether can only be reeled out during T_1 and only reeled in during T_2, resulting in the desired phase pattern. As for the phase fixing parameter, $p_{\text{fix}} = 0.7$ has been found to work well. It should be noted that t_{f} is still a free decision variable of the OCP. Only the ratio of reel-in to reel-out phase gets fixed beforehand with the parameter p_{fix}.

1.3. Emergency Scenarios for Airborne Wind Energy Systems

It is clear that a lot can go wrong with a fully autonomous multi-kite AWES. Even with the most careful implementation of all components, there are still a multitude of uncontrollable variables that need to be taken into account, e.g. bird strike, bad weather conditions, sabotage or careless amateur pilots. On top of that, there are also all the regular risks of malfunction for mechatronic systems. Since the AWES is designed to be fully autonomous, all possible scenarios must be thought of in advance in order to prepare a suitable countermeasure. The most straightforward emergency responses are some sort of an emergency landing in case the AWES is damaged and needs repairs. In the case of minor malfunctions such as one of several redundant sensors breaking down or the on-board battery losing some of its maximum capacity over time, there needs to be a judgment call as to whether the AWES has to land or not. Here of course, the modus operandi must be to try to land *before* any serious malfunction occurs that would impede the control and maneuverability of the AWES, since a landing with nominal control and sensor functionality is always safer and more efficient than an emergency landing, no matter how carefully the latter is planned. In addition to a complete emergency landing, there are some scenarios for which it would be sufficient to abandon the high altitude pumping trajectory for a more conservative but still efficient pumping trajectory in lower altitudes. The most common of those scenarios would be unfavorable weather conditions that make it unsafe to continue to fly the initial pumping trajectory, but do not pose any threat to the AWES in the lower altitude pumping trajectory. Thus, it is possible to avoid the time and energy expensive landing and starting procedures.

On the one hand, multi-kite AWESs are much more prone to emergencies than single kite systems, simply because they have two or more kites as opposed to one, allowing for a higher number of mechatronic parts that can potentially break or get damaged. In addition, a multi-kite system requires a more sophisticated control setup that will be naturally less robust and more error prone than its simpler counterparts. Even when using the same control setup as for single kites, a multi-kite system needs at least one additional controller per kite, adding to the complexity of the overall control structure. On the other hand, the increased complexity of the multi-kite AWES can be potentially beneficial in cases of emergency. One kite with broken or otherwise compromised actuation could e.g. be indirectly supported by the other kites by compensating for some of its forces. One can even imagine a scenario where a group of kites carries the entire weight of another broken kite to the ground.

1.4. Related Work

Since the multi-kite AWES is intended to be controlled completely autonomous, there are no experienced pilots who can use their skill and experience to cope with these emergency scenarios. Hence, it is of importance to think of possible emergencies and prepare the controller's actions for these cases. This problem however is not so relevant for the conceptual and prototyping stages of an AWES development process. Simulations are inherently non-dangerous and for prototypes, a switch to manual control can give an experienced pilot access to the AWES's actuation in case something goes wrong. Thus, there exists no detailed research yet in the field of emergency trajectories for AWESs.

The work done on trajectory optimization for AWESs focuses almost exclusively on the generation of suitable power cycles [21, 10, 19, 45, 24, 26]. A notable exception is the computation of a transition from one trajectory to another for a single-kite system by Horn [19]. In terms of landing trajectories, one should note the landing trajectory of a single kite computed by Koenemann [23]. Within an OCP framework, he simplifies the system by restricting the dynamics to a two dimensional space and introduces a quasi static tether model that accounts for tether sag, but neglects tether vibrations. While some of the OCP formulation can also be used for dual kites, the two dimensional approximation of the kite dynamics only makes sense in the final stages of a single kite landing.

In the field of trajectory planning and optimization for passenger aircraft, emergency landing trajectories play an important role. While some of them employ an optimal control (OC) or dynamic programming (DP) framework [30, 42], others use different kinds of path primitives to build trajectories [38, 1, 5]. In the literature concerning unmanned aerial vehicles (UAV), the methodology is very similar. While some authors employ an OC framework, most rely on path primitives like Dubins curves to put together landing trajectories [43, 29, 12, 13]. A number of papers also concern themselves with decision making problems such as choosing a suitable landing site or proposing logical conditions that trigger an emergency landing procedure [37, 29]. The results of the papers related to passenger aircraft, however, can hardly be adapted for AWESs for the following reasons:

- Most papers about emergency landing trajectories for passenger aircraft consider the scenario of an engine failure that results in zero thrust while the rest of the actuation stays intact. Multi-kite AWESs rely on a different control concept that relies on actuation via the main tether and the kites' flaps.

- The motion of a single aircraft is well understood. This makes it possible to derive feasible solutions in an analytical fashion using e.g. Dubins curves. These analytical solutions can then be used as initial guesses for optimization problems or as motion primitives for other planning algorithms. The motion of multi-kite AWES's is not nearly as well understood.

- The fact that there are at least two kites in a multi-kite AWES makes it possible for the kites to indirectly control each other, especially in case of compromised

actuation on one of the kites. This is a dimension of the problem that is completely new.

- The most common flight maneuver of a passenger aircraft is steady level longitudinal flight, which is very useful as an initial state for a trajectory. Given that the most common flight maneuver for a multi-kite AWES is flying crosswind-maneuvers, an emergency trajectory will generally start on a power cycle. This is more difficult conceptually and numerically, since there are a multitude of different initial conditions to consider inside one power cycle that all differ a lot from each other.

Considering all this, it seems prudent to choose an OC framework for the work of this thesis. It is a very flexible framework for, which enables the engineer to adapt to a multitude of different models and circumstances relatively easily by editing the OCP's constraints or cost function. It is possible to find automated, numerical solutions to an OCP, which do not require as much analytical insight into the problem as an analytical solution. This is important since it enables us to produce solution procedures that are robust with regards to a change in the modeling of the AWES or its environment. An analytical approach would make it necessary to re-derive the analytical formulas for every change in the problem's model, e.g. a derivation of different motion primitives for every new multi-kite configuration. This approach makes sense for a fixed emergency scenario like the emergency landing of a passenger aircraft, but not when facing varying emergency scenarios and AWES models. Additionally, an OCP framework is used to solve not only for any feasible trajectory, but the optimal feasible trajectory. For example, one can compute the feasible trajectory with the smallest control input or the smallest terminal velocity. Hence, we generally expect to get better results using OC than other approaches that are not based on optimization.

1.5. Organization of Contents

The aim of this thesis is the construction of a general and modular OC framework that can be used to compute and analyze a variety of multi-kite emergency trajectories. This includes an emergency response strategy as well as a homotopy strategy for solving the OCPs within the framework. The framework is conceptualized in a modular fashion such that it can be easily expanded by adding more complex models or new emergency scenarios. After computing a number of different optimal emergency trajectories, we will explain why these particular trajectories are optimal and how the optimal trajectories change with different models and parameters. We also want to find out how badly a multi-kite system can be compromised before it is impossible to perform an emergency landing and whether it is possible for a compromised kite to be indirectly controlled by the rest of the AWES.

We will start off in chapter 2 with a summary of all the theoretical background and the methods used. The first section of this chapter will discuss the modeling approach

used to model the AWES. We then go on to explain the direct collocation scheme with which the OCPs in this thesis are discretized. The next section gives an overview of the interior-point method of the numerical solver. The last section of chapter 2 explains the homotopy strategy that is employed to solve the OCPs in this thesis.

Chapter 3 concerns itself with the layout of an emergency response concept of a multi-kite AWES. It first goes into detail about the nature of the expected emergency scenarios. Then we present a state-machine that can be used for the detection of emergency scenarios. Lastly, we introduce a hierarchical emergency landing strategy in order to handle different kinds of emergency scenarios in one unified framework.

Chapter 4 describes how the AWES is modeled. This is done by first listing the system's states, controls and coordinate system. We then elaborate on the Lagrangian mechanics of the system and explain which wind and atmospheric model is used. Next, we discuss which generalized forces are acting on the system. The chapter is concluded with an explanation of how the kites' on-board batteries are modeled.

Chapter 5 introduces a modular framework of building blocks that make up every OCP in this thesis. First we list the different components of the cost functions, then we go on to the equality and inequality constraints. This is followed by a catalog of all the homotopy steps that can be put together to a full homotopy procedure. Afterwards, we discuss how the initial guess for the first homotopy step is generated and give an account of the different numerical issues that have come up during the implementation of the OCPs.

Chapter 6 explains how the problem formulations discussed in this thesis are implemented into the `python` toolbox `AWEbox`. After an overview of `AWEbox`'s functionalities, we give an account of the contributions made to `AWEbox` during the course of this thesis. Section 6.2 will then briefly elaborate on the `CasADi` toolbox, which is used within `AWEbox`, and its numerical backend `IPOPT`.

Chapter 7 takes the framework introduced in chapter 5 and applies it to landing trajectories with nominal flight behavior. After stating the OCP formulation and homotopy schedule of the problem, the corresponding solutions are analyzed. This includes a discussion of the solutions of various homotopy steps, a study of the solution's sensitivity to a change in the reference wind speed, as well some insights in how the potential and kinetic energy of the system behave during the landing.

Chapter 8 then gives an overview of all the emergency trajectories and their corresponding OCP formulations that have been implemented over the course of this thesis besides the nominal landing. These include transition trajectories as well as landing trajectories with compromised flight behavior. The latter comprises three different emergency scenarios, specifically an actuator malfunction, a broken on-board battery and structural damage.

Chapter 9 then concludes the thesis by giving a concise assessment of the results and an outlook onto future research in the field of optimal control for multi-kite emergency trajectories.

2. Optimal Control Methods and Theory

This chapter gives a summary of all the theoretical background and methods that are relevant to this thesis. Section 2.1 gives an overview of the Lagrange modeling approach that is used to represent the AWES as an index 1 differential algebraic equation (DAE). Section 2.2 then explains how the OCPs that are solved in the following chapters are discretized using direct collocation. Section 2.3 elaborates on how interior-point (IP) methods are used to numerically solve the discretized OCP and why an IP method is chosen over a sequential quadratic programming (SQP) method. Lastly, section 2.4 goes into detail about the homotopy strategy that is used to solve the OCPs in this thesis.

2.1. System Modeling with DAEs

The modeling of physical systems like the multi-kite AWES is typically done using a Lagrange approach [17, 45] because it provides a framework that can be easily generalized and automated. To do so, we define $t \in [0, t_\mathrm{f}]$ to be the time interval in which the system is evolving with final time t_f. We employ the generalized coordinates $\mathbf{q}(t) \in \mathcal{Q}$, which are a set of independent variables that can describe any possible configuration of the physical system. These configurations can be represented as a manifold $\mathcal{M}$ defined by the invariant $\mathbf{g}(\mathbf{q}, t) = \mathbf{0}$, where $\mathcal{M} = \{\mathbf{q} \in \mathcal{Q} \mid \mathbf{g}(\mathbf{q}(t), t) = \mathbf{0}, t \in [0, t_\mathrm{f}]\}$ is the set of all possible configurations of the physical system. The invariant $\mathbf{g}$ is a map from $\mathcal{Q} \times [0, t_\mathrm{f}]$ to $\mathcal{G} \in \mathbb{R}^{m_\mathbf{g} \times n_\mathbf{g}}$. The dimensions $m_\mathbf{g}$ and $n_\mathbf{g}$ are usually those of a vector, but can be arbitrary.

With the help of $\mathbf{q}$ and $\mathbf{g}(\mathbf{q}, t)$ we can formulate the Lagrange function

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\lambda}) = T(\dot{\mathbf{q}}) - V(\mathbf{q}) - \boldsymbol{\lambda}^\mathsf{T} \mathbf{g}(\mathbf{q}, t), \tag{2.1}$$

where T is the system's kinetic energy, V the system's potential energy and $\boldsymbol{\lambda}$ the vector of Lagrange multipliers. From there, the Lagrange dynamics

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{F}, \quad \mathbf{g}(\mathbf{q}, t) = \mathbf{0} \tag{2.2}$$

are derived, with the non-conservative generalized forces $\mathbf{F}$. The non-conservative generalized forces are defined by the principle of virtual work: For an infinitesimal displacement $\delta\mathbf{q}$ in the generalized coordinates, the work done on the system by the forces $\mathbf{F}$

is

$$\delta W = \langle \delta \mathbf{q}, \mathbf{F} \rangle, \tag{2.3}$$

where $\langle \cdot, \cdot \rangle$ is the scalar product on $\mathcal{Q}$. In the usual case, where $\mathcal{Q} \in \mathbb{R}^{n_\mathbf{q}}$, $\mathbf{F} \in \mathbb{R}^{n_\mathbf{q}}$ is a projection of the non-conservative forces acting on the system on $\mathcal{Q}$ and the scalar product is simply defined as $\delta \mathbf{q}^\mathsf{T} \mathbf{F}$.

Equation (2.2) is an index 3 DAE. Within an OC context, DAEs are usually considered as root-finding problems. The Jacobian of an index 3 DAE is non-singular, meaning that we cannot use a Newton-type solver for the corresponding root-finding problem. Hence, high-index DAEs are best treated by performing index reduction [17]. Using index reduction, (2.2) can be turned into an index 1 DAE that has an invertible Jacobian and can therefore be treated with Newton-type solvers. To do so, we first compute the Lagrange dynamics

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} &= \frac{\mathrm{d}}{\mathrm{d}t}(T_{\dot{\mathbf{q}}} - V_{\dot{\mathbf{q}}} - (\boldsymbol{\lambda}^\mathsf{T}\mathbf{g})_{\dot{\mathbf{q}}}) - T_\mathbf{q} + V_\mathbf{q} + (\boldsymbol{\lambda}^\mathsf{T}\mathbf{g})_\mathbf{q} \\ &= T_{\dot{\mathbf{q}}t} + T_{\dot{\mathbf{q}}\dot{\mathbf{q}}}\ddot{\mathbf{q}} + V_\mathbf{q} + (\boldsymbol{\lambda}^\mathsf{T}\mathbf{g})_\mathbf{q} \\ &= T_{\dot{\mathbf{q}}\dot{\mathbf{q}}}\ddot{\mathbf{q}} + V_\mathbf{q} + (\boldsymbol{\lambda}^\mathsf{T}\mathbf{g})_\mathbf{q} = \mathbf{F}, \end{aligned} \tag{2.4}$$

where we employ the shorthand notation of $\frac{\partial}{\partial \mathbf{q}}(\cdot) = (\cdot)_\mathbf{q}$. We also employ the knowledge that $\mathbf{g}(\mathbf{q})_{\dot{\mathbf{q}}} = T(\dot{\mathbf{q}})_\mathbf{q} = V(\mathbf{q})_{\dot{\mathbf{q}}} = 0$ and that $T(\dot{\mathbf{q}})_{\dot{\mathbf{q}}t} = 0$ because the kinetic energy does not explicitly depend on the time t.

Furthermore, we compute the first and second derivative of $\mathbf{g}(\mathbf{q}, t)$ with respect to time and then set $\ddot{\mathbf{g}}$ to zero, resulting in

$$\begin{aligned} \dot{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}, t) &= \mathbf{g}_t + \mathbf{g}_\mathbf{q}\dot{\mathbf{q}} \\ \ddot{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, t) &= \dot{\mathbf{g}}_t + \dot{\mathbf{g}}_\mathbf{q}\dot{\mathbf{q}} + \mathbf{g}_\mathbf{q}\ddot{\mathbf{q}} = \mathbf{0}. \end{aligned} \tag{2.5}$$

Note that $\mathbf{g}_t$ is only the partial derivative with respect to time, while $\dot{\mathbf{g}}$ denotes the total derivative.

Equations (2.4) and (2.5) can be written as the system

$$\underbrace{\begin{bmatrix} T_{\dot{\mathbf{q}}\dot{\mathbf{q}}} & \mathbf{g}_\mathbf{q}^\mathsf{T} \\ \mathbf{g}_\mathbf{q} & \mathbf{0} \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{F} - V_\mathbf{q} \\ -\dot{\mathbf{g}}_\mathbf{q}\dot{\mathbf{q}} - \mathbf{g}_\mathbf{q}\ddot{\mathbf{q}} \end{bmatrix}, \tag{2.6}$$

which, together with the consistency conditions $\mathbf{g}(\mathbf{q}(0), 0) = \mathbf{0}$ and $\dot{\mathbf{g}}(\mathbf{q}(0), \dot{\mathbf{q}}(0), 0) = \mathbf{0}$, is equivalent to (2.2). Note that (2.6) is linear in $[\ddot{\mathbf{q}}^\mathsf{T}, \boldsymbol{\lambda}^\mathsf{T}]^\mathsf{T}$. Hence, it seems tempting to invert the mass matrix $\mathbf{M}$ of (2.6) to get an explicit formulation for $[\ddot{\mathbf{q}}^\mathsf{T}, \boldsymbol{\lambda}^\mathsf{T}]^\mathsf{T}$ that can then be integrated with an explicit integration scheme. However, inverting $\mathbf{M}$ would destroy some of the natural sparsity, making a treatment with explicit integrators computationally more expensive. Also, DAEs are better treated with implicit integrators since they are usually stiff. A DAE is called stiff, if some of its Jacobian's eigenvalues have a very

large negative real part. These very stable modes can be involuntarily excited by the explicit integrator, leading to unstable behavior in the numerical integration. Hence, we use an implicit integration scheme, namely the direct collocation method explained in section 2.2.

When solving (2.6), $\ddot{\mathbf{g}}(t) = \mathbf{0}$ gets integrated implicitly with the corresponding initial values ensuring that $\mathbf{g}(t) = \mathbf{0}$ holds for all times t. This is however only valid if (2.6) is solved exactly. When using numerical approaches to solve (2.6), the integral of $\ddot{\mathbf{g}}$ is only approximated. This can lead to a numerical drift, especially for long integration intervals, resulting in $\mathbf{g}(t) \neq \mathbf{0}$ for some t. To counteract the numerical drift, we employ Baumgarte stabilization [17]. Instead of enforcing $\ddot{\mathbf{g}}(t) = \mathbf{0}$, we set

$$\ddot{\mathbf{g}}(t) = -2\kappa\dot{\mathbf{g}}(t) - \kappa^2\mathbf{g}(t). \tag{2.7}$$

This turns the dynamics of the consistency constraints into a stable oscillator for $\kappa \in \mathbb{R}_+$, with a double eigenvalue at $-\kappa \in \mathbb{C}_-$. Since we fixed $\mathbf{g}(0) = \mathbf{0}$ and $\dot{\mathbf{g}}(0) = \mathbf{0}$, $\mathbf{g}(t)$ is stable around $\mathbf{g}(t) = \mathbf{0}$, counteracting the numerical drift. For a more compact notation, we introduce

$$\mathbf{c}_{\text{inv}}(0) = [\mathbf{g}(0), \dot{\mathbf{g}}(0)]^\mathsf{T} = \mathbf{0} \tag{2.8}$$

as the consistency constraint.

2.2. Direct Collocation

The OCP we want to approximate can be written as

$$\begin{aligned}
\underset{\mathbf{x}(\cdot),\,\mathbf{u}(\cdot),\,\boldsymbol{\lambda}(\cdot),\,t_\text{f},\,\mathbf{p}}{\text{minimize}} \quad & J(\mathbf{x}(\cdot), \mathbf{u}(\cdot), \boldsymbol{\lambda}(\cdot), t_\text{f}, \mathbf{p}) \\
\text{subject to} \quad & \mathbf{0} = \mathbf{f}(\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), \mathbf{p}), && \forall t \in [0, t_\text{f}], \\
& \mathbf{0} = \mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), \mathbf{p}), && \forall t \in [0, t_\text{f}], \\
& \mathbf{0} = \mathbf{c}_0(\mathbf{x}(0), \mathbf{u}(0), \boldsymbol{\lambda}(0), \mathbf{p}), \\
& \mathbf{0} = \mathbf{c}_\text{f}(\mathbf{x}(t_\text{f}), \mathbf{u}(t_\text{f}), \boldsymbol{\lambda}(t_\text{f}), \mathbf{p}), \\
& \mathbf{0} \leq \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), \mathbf{p}), && \forall t \in [0, t_\text{f}], \\
& \mathbf{0} \leq \mathbf{h}_0(\mathbf{x}(0), \mathbf{u}(0), \boldsymbol{\lambda}(0), \mathbf{p}), \\
& \mathbf{0} \leq \mathbf{h}_\text{f}(\mathbf{x}(t_\text{f}), \mathbf{u}(t_\text{f}), \boldsymbol{\lambda}(t_\text{f}), \mathbf{p}), \\
& \mathbf{x}_{\min}(\mathbf{p}) \leq \mathbf{x}(t) \leq \mathbf{x}_{\max}(\mathbf{p}), && \forall t \in [0, t_\text{f}], \\
& \mathbf{u}_{\min}(\mathbf{p}) \leq \mathbf{u}(t) \leq \mathbf{u}_{\max}(\mathbf{p}), && \forall t \in [0, t_\text{f}], \\
& \boldsymbol{\lambda}_{\min} \leq \boldsymbol{\lambda}(t) \leq \boldsymbol{\lambda}_{\max}, && \forall t \in [0, t_\text{f}], \\
& 0 \leq t_\text{f}.
\end{aligned} \tag{2.9}$$

In this generic OCP, the decision variables are the system states $\mathbf{x}(\cdot) \in \mathbb{R}^{n_\mathbf{x}}$, the controls $\mathbf{u}(\cdot) \in \mathbb{R}^{n_\mathbf{u}}$, the algebraic variables $\boldsymbol{\lambda}(\cdot) \in \mathbb{R}^{n_{\boldsymbol{\lambda}}}$, the final time $t_\text{f} \in \mathbb{R}_+$ and a

parameter vector $\mathbf{p}$[1]. We use the $(\cdot)$-notation to stress that the decision variables of the OCP are functions and that J is a functional. The cost functional usually consists of a Lagrange term $L_\mathrm{T} = \int_0^{t_\mathrm{f}} l(t)\ \mathrm{d}t$ and a Mayer term $\phi(t_\mathrm{f})$ such as

$$J(\mathbf{x}(\cdot), \mathbf{u}(\cdot), \boldsymbol{\lambda}(\cdot), t_\mathrm{f}, \mathbf{p}) = \int_0^{t_\mathrm{f}} l(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), \mathbf{p})\ \mathrm{d}t + \phi(\mathbf{x}(t_\mathrm{f}), \boldsymbol{\lambda}(t_\mathrm{f}), t_\mathrm{f}, \mathbf{p}). \tag{2.10}$$

The Lagrange term is an integral term that depends on the decision values at all $t \in [0, t_\mathrm{f}]$ and is therefore also called running cost. The Mayer term, or final cost, only depends on the decision values at the final time t_f. The implicit DAE

$$\mathbf{0} = \mathbf{f}(\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), \mathbf{p}) \tag{2.11}$$

describes the system's dynamics as explained in section 2.1. Adding it to the OCP constrains the problem to physical solutions. Furthermore, the decision variables are constrained by their respective bounds as well as the path constraints $\mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), \mathbf{p})$, their initial equality constraints $\mathbf{c}_0(\mathbf{x}(0), \mathbf{u}(0), \boldsymbol{\lambda}(0), \mathbf{p})$ and their terminal equality constraints $\mathbf{c}_\mathrm{f}(\mathbf{x}(t_\mathrm{f}), \mathbf{u}(t_\mathrm{f}), \boldsymbol{\lambda}(t_\mathrm{f}), \mathbf{p})$. In case of the index 1 DAE described in section 2.1, part of the initial equality constraints would be the consistency constraint $\mathbf{c}_\mathrm{inv}(0) = 0$.

In section 2.1, the DAE describing the system dynamics was derived using the generalized coordinates $\mathbf{q}$. Hence, (2.6) can be written as the implicit function

$$\mathbf{f}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t), \boldsymbol{\lambda}(t), t, \mathbf{p}) = \mathbf{0}. \tag{2.12}$$

Depending on how $\mathbf{q}$, $\mathbf{x}$, $\mathbf{u}$ and $\boldsymbol{\lambda}$ are chosen, one can define a coordinate transformation

$$T_\mathrm{c} : \left[\mathbf{q}^\mathsf{T}, \dot{\mathbf{q}}^\mathsf{T}, \ddot{\mathbf{q}}^\mathsf{T}, t\right]^\mathsf{T} \mapsto \left[\mathbf{x}^\mathsf{T}, \dot{\mathbf{x}}^\mathsf{T}, \mathbf{u}^\mathsf{T}\right]^\mathsf{T} \tag{2.13}$$

in order to arrive at (2.11).

The inequality constraints on $\mathbf{x}(t)$, $\mathbf{u}(t)$ and $\boldsymbol{\lambda}(t)$ can also be represented as $\mathbf{x}(t) \in \mathcal{X}_\mathrm{B}$, $\mathbf{u}(t) \in \mathcal{U}_\mathrm{B}$ and $\boldsymbol{\lambda}(t) \in \boldsymbol{\lambda}_\mathrm{B}$ where $\mathcal{X}_\mathrm{B}$, $\mathcal{U}_\mathrm{B}$ and $\boldsymbol{\lambda}_\mathrm{B}$ are usually chosen to be convex sets. We explicitly permit the bounds on $\mathbf{x}(t)$ and $\mathbf{u}(t)$ to depend on $\mathbf{p}$. The inequality constraint $\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), \mathbf{p}) \leq \mathbf{0}$ permits more complex types of inequality constraints than box or set constraints. It should be noted that (2.9) is in general not a convex problem. Even if we choose the cost functional J, the inequality constraints $\mathbf{h}$ and the equality constraints $\mathbf{c}$ as well as the sets $\mathcal{X}_\mathrm{B}$, $\mathcal{U}_\mathrm{B}$ and $\boldsymbol{\lambda}_\mathrm{B}$ to be convex, the DAE will usually constrain the optimization variables to a non-convex set, rendering the whole problem non-convex. This stays the same also after discretizing the problem.

Since (2.9) is generally non-convex, the initial guess does not only determine how fast a minimum is found, but also which local minimum is found. There is no guarantee that

[1]Technically, $\mathbf{p}$ is a variable vector since it is part of the optimization variables. However, we continue to call it parameter vector to distinguish it from the other optimization variables.

the global minimum can be found and even if it is found, there is no way of checking that it truly is the global minimum. This is because any numerical NLP solver only uses local information, from which it cannot deduce global properties of the NLP, except when it is convex.

As mentioned in section 2.1, we choose an implicit integration scheme to handle (2.11). The continuous OCP (2.9) is an infinite dimensional optimization problem since $t \in [0, t_\mathrm{f}]$ runs trough an infinite amount of values, and thus results in an infinite amount of optimization variables and constraints. Although it is possible to solve this problem with an indirect approach using Pontryagin's maximum principle, this is not viable for the complex nonlinear systems that multi-kite AWESs are, especially since every change in the model would require a new derivation of the OCP's solution by hand.

Thus, we pursue the strategy of "first discretize, then optimize", leading to the method of direct collocation that has already produced good results for similar problems in the past [19]. Also, direct collocation has a larger region of convergence compared to e.g. a multiple shooting approach [16]. Specifically, we choose direct collocation of type Radau IIA due to its good stability properties (A- and L-stability) [32]. The main idea of direct collocation is to divide the solution's time axis into multiple collocation intervals and to use a different polynomial on each interval to approximate the system's DAE.

The first step to finding an approximate solution to (2.9) is discretization. The time interval $[0, t_\mathrm{f}]$ is divided into m collocation intervals $t_i = [t_{i,0}, t_{i,\mathrm{f}}]$, where $t_{i,0}$ and $t_{i,\mathrm{f}}$ are the first and last point on each collocation interval and $i \in \{0, 1, .., m-1\} = \mathcal{I}$. The intervals are set up such that $\cup_{i=0}^{m-1} t_i = [0, t_\mathrm{f}]$ and $\cap(t_i, t_{i+1}) = t_{i,\mathrm{f}} = t_{i+1,0}$. The normalized time on each collocation interval t_i is $\bar{\tau}_i \in [0, 1]$, such that $t = t_{i,0} + \bar{\tau}_i \frac{t_\mathrm{f}}{m}$.

Each collocation interval t_i has $n+1$ control points $t_{i,j}$ with $j \in \{0, .., n\} = \mathcal{J}$. The discretized control $\mathbf{u}_i$ is piecewise constant in each collocation interval. Choosing a control parameterization with local support ensures that the sparsity of the OCP is preserved when discretizing it. The control $\mathbf{u}_i$ is kept constant within each collocation interval to avoid discontinuities that would destroy the accuracy of the integration scheme [16].

On collocation interval t_i we refer to $t_{i,j}$ in terms of the normalized time $\bar{\tau}_j$ for ease of notation. In the following, it will also be handy to define the sets $\mathcal{I}^- = \{0, ..., m-2\}$ and $\mathcal{I}_- = \{1, ..., m-1\}$. The sets $\mathcal{J}^-$ and $\mathcal{J}_-$ are defined analogously as $\mathcal{J}^- = \{0, ..., n-1\}$ and $\mathcal{J}_- = \{1, ..., n\}$. Furthermore, we define $\bar{m} = m - 1$.

On each collocation interval we define a Lagrange polynomial of degree n as

$$\bar{\mathbf{x}}_i(\bar{\tau}) = \sum_{j=0}^{n} \xi_j(\bar{\tau}) \mathbf{x}_{i,j}, \tag{2.14}$$

where

$$\xi_j(\bar{\tau}) = \prod_{k=0, k\neq j}^{n} \frac{\bar{\tau}_k - \bar{\tau}}{\bar{\tau}_k - \bar{\tau}_j}. \tag{2.15}$$

Hence, the Lagrange polynomial has the property that $\bar{\mathbf{x}}_i(\bar{\tau}_j) = \mathbf{x}_{i,j}$. The notation $\mathbf{x}_{i,j}$ refers to the system states at the control point $t_{i,j}$, however only in terms of the discretized

OCP. Thus, $\mathbf{x}_{i,j}$ is only an approximation of the real solution $\mathbf{x}(t_{i,j})$. The Lagrange polynomial (2.14) can be differentiated with respect to time to yield

$$\dot{\bar{\mathbf{x}}}_i(\bar{\tau}) = \sum_{j=0}^{n} \frac{\partial \xi_j(\bar{\tau})}{\partial \bar{\tau}} \frac{\mathrm{d}\bar{\tau}}{\mathrm{d}t} \mathbf{x}_{i,j} = \sum_{j=0}^{n} \frac{\partial \xi_j(\bar{\tau})}{\partial \bar{\tau}} \frac{m}{t_\mathrm{f}} \mathbf{x}_{i,j}. \tag{2.16}$$

We choose $t_{i,j}$ to be Radau collocation points, meaning that the end of each collocation interval is a collocation point, but the beginning is not. Thus, the last collocation point of interval t_i also lies in t_{i+1} but is only a collocation point of t_i. The notation $t_{i,0}$ is used for the beginning of each collocation interval, even though it is not a collocation point. At each collocation point, the Lagrange polynomial $\bar{\mathbf{x}}_i(\bar{\tau})$ must satisfy the DAE of (2.9). Hence, we have

$$\mathbf{0} = \mathbf{f}(\bar{\mathbf{x}}_i(\bar{\tau}_j), \dot{\bar{\mathbf{x}}}_i(\bar{\tau}_j), \mathbf{u}_i, \boldsymbol{\lambda}_{i,j}, t_\mathrm{f}, \mathbf{p}), \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}_-. \tag{2.17}$$

The indexing convention for $\boldsymbol{\lambda}_{i,j}$ is the same as for $\mathbf{x}_{i,j}$. To ensure that the solution approximated with the Lagrange polynomials $\bar{\mathbf{x}}_i$ is continuous, we need to enforce

$$\bar{\mathbf{x}}_i(1) = \mathbf{x}_{i+1,0}, \qquad \forall i \in \mathcal{I}^-. \tag{2.18}$$

Since we use Radau collocation points, $\bar{\tau}_n = 1$ and the last collocation point coincides with the end of the interval t_i. Thus, there is no need to evaluate $\bar{\mathbf{x}}_i(1)$ and (2.18) turns into

$$\mathbf{x}_{i,n} = \mathbf{x}_{i+1,0}, \qquad \forall i \in \mathcal{I}^-. \tag{2.19}$$

Equations (2.17) and (2.19) are called collocation constraints and continuity constraints. They are further referred to as

$$\mathbf{0} = \mathbf{c}_\mathrm{col}(\mathbf{x}_{i,j}, \boldsymbol{\lambda}_{i,j}, \mathbf{u}_i, t_\mathrm{f}, \mathbf{p}), \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}_- \tag{2.20}$$

$$\mathbf{0} = \mathbf{c}_\mathrm{con}(\mathbf{x}_{i,n}, \mathbf{x}_{i+1,0}, \mathbf{p}), \qquad \forall i \in \mathcal{I}^-. \tag{2.21}$$

With all this we can approximate (2.9) with

$$
\begin{aligned}
\underset{\mathbf{x},\mathbf{u},\boldsymbol{\lambda},t_\mathrm{f},\mathbf{p}}{\text{minimize}} \quad & \bar{J}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda},t_\mathrm{f},\mathbf{p}) \\
\text{subject to} \quad & \mathbf{0} = \mathbf{c}_\mathrm{col}(\mathbf{x}_{i,j},\boldsymbol{\lambda}_{i,j},\mathbf{u}_i,t_\mathrm{f},\mathbf{p}), \quad \forall i \in \mathcal{I}, j \in \mathcal{J}_-, \\
& \mathbf{0} = \mathbf{c}_\mathrm{con}(\mathbf{x}_{i,n},\mathbf{x}_{i+1,0},\mathbf{p}), \quad \forall i \in \mathcal{I}^-, \\
& \mathbf{0} = \mathbf{c}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda},\mathbf{p}), \\
& \mathbf{0} = \mathbf{c}_0(\mathbf{x}_{0,0},\mathbf{u}_0,\boldsymbol{\lambda}_{0,0}), \\
& \mathbf{0} = \mathbf{c}_\mathrm{f}(\mathbf{x}_{\bar{m},n},\mathbf{u}_{\bar{m}},\boldsymbol{\lambda}_{\bar{m},n},\mathbf{p}), \\
& \mathbf{0} \leq \mathbf{h}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda},\mathbf{p}), \\
& \mathbf{0} \leq \mathbf{h}_0(\mathbf{x}_{0,0},\mathbf{u}_0,\boldsymbol{\lambda}_{0,0},\mathbf{p}), \\
& \mathbf{0} \leq \mathbf{h}_\mathrm{f}(\mathbf{x}_{\bar{m},n},\mathbf{u}_{\bar{m}},\boldsymbol{\lambda}_{\bar{m},n},\mathbf{p}), \\
& \mathbf{x}_\mathrm{min}(\mathbf{p}) \leq \mathbf{x} \leq \mathbf{x}_\mathrm{max}(\mathbf{p}), \\
& \mathbf{u}_\mathrm{min}(\mathbf{p}) \leq \mathbf{u} \leq \mathbf{u}_\mathrm{max}(\mathbf{p}), \\
& \boldsymbol{\lambda}_\mathrm{min} \leq \boldsymbol{\lambda} \leq \boldsymbol{\lambda}_\mathrm{max}, \\
& 0 \leq t_\mathrm{f},
\end{aligned}
\tag{2.22}
$$

where we use the notation

$$\mathbf{x} = [\mathbf{x}_{0,0}, ..., \mathbf{x}_{\bar{m},n}]^\mathsf{T} \tag{2.23}$$

$$\mathbf{u} = [\mathbf{u}_0, ..., \mathbf{u}_{\bar{m}}]^\mathsf{T} \tag{2.24}$$

$$\boldsymbol{\lambda} = [\boldsymbol{\lambda}_{0,1}, ..., \boldsymbol{\lambda}_{1,1}, ..., \boldsymbol{\lambda}_{\bar{m},n}]^\mathsf{T}. \tag{2.25}$$

Keep in mind that that the algebraic variables $\boldsymbol{\lambda}$ are not defined for all control points, but only for the collocation points.

The cost functional is approximated as

$$\bar{J}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda},t_\mathrm{f},\mathbf{p}) = \underbrace{\frac{t_\mathrm{f}}{m}\sum_{i=0}^{\bar{m}}\sum_{j=1}^{n} w_{\mathrm{q},j}\, l(\mathbf{x}_{i,j},\mathbf{u}_i,\boldsymbol{\lambda}_{i,j},\mathbf{p})}_{\bar{L}_\mathrm{T}} + \phi(\mathbf{x}_{\bar{m},n},\boldsymbol{\lambda}_{\bar{m},n},t_\mathrm{f},\mathbf{p}), \tag{2.26}$$

with the quadrature weights $\mathbf{w}_\mathrm{q} = [w_{\mathrm{q},1}, ..., w_{\mathrm{q},n}]$.

To derive the approximation for the Lagrange term $L_\mathrm{T} = \int_0^{t_\mathrm{f}} l(t)\ \mathrm{d}t$, we follow [19]. First, we want to approximate the Lagrange term $L_{\mathrm{T},i} = \int_{t_{i,0}}^{t_{i,\mathrm{f}}} l(t)\ \mathrm{d}t \approx \bar{L}_{\mathrm{T},i}$ under the assumption that $\bar{L}_{\mathrm{T},i}(t_{i,0}) = 0$. To do so, we write out (2.16) for all collocation nodes of the collocation interval i for the derivative l_i as

$$\begin{bmatrix} l_i(\bar{\tau}_1) \\ \vdots \\ l_i(\bar{\tau}_m) \end{bmatrix} = \frac{m}{t_\mathrm{f}} \begin{bmatrix} \xi_1'(\bar{\tau}_1) & \cdots & \xi_n'(\bar{\tau}_1) \\ \vdots & \ddots & \vdots \\ \xi_1'(\bar{\tau}_n) & \cdots & \xi_n'(\bar{\tau}_n) \end{bmatrix} \begin{bmatrix} \bar{L}_{\mathrm{T},i}(\bar{\tau}_1) \\ \vdots \\ \bar{L}_{\mathrm{T},i}(\bar{\tau}_n) \end{bmatrix}, \tag{2.27}$$

where we use the notation $\frac{\partial \xi_i}{\partial \bar{\tau}} = \xi_j'$. When solving this equation for $\left[L_{\bar{\mathrm{T}},i}(\bar{\tau}_1), \cdots, L_{\bar{\mathrm{T}},i}(\bar{\tau}_n)\right]^\mathsf{T}$ and plugging the solution into (2.19), we arrive at

$$\bar{L}_{\mathrm{T},i}(1) = [\xi_1(1), ..., \xi_2(1)] \frac{t_\mathrm{f}}{m} \begin{bmatrix} \xi_1'(\bar{\tau}_1) & \cdots & \xi_n'(\bar{\tau}_1) \\ \vdots & \ddots & \vdots \\ \xi_1'(\bar{\tau}_n) & \cdots & \xi_n'(\bar{\tau}_n) \end{bmatrix}^{-1} \begin{bmatrix} l_i(\bar{\tau}_1) \\ \vdots \\ l_i(\bar{\tau}_n) \end{bmatrix}. \tag{2.28}$$

Note that $[\xi_1(1), \cdots, \xi_2(1)] = [0, \cdots, 0, 1]$ since we are using Radau collocation points. Since we have assumed that $\bar{L}_{\mathrm{T},i}(t_{i,0}) = 0$ for all collocation intervals i, we can sum up the intervals as

$$\bar{L}_\mathrm{T} = \sum_{i=0}^{\bar{m}} \bar{L}_{\mathrm{T},i}, \tag{2.29}$$

which is equivalent to the expression used in equation (2.26) for the quadrature weights

$$\mathbf{w}_\mathrm{q} = [\xi_1(1), \cdots, \xi_2(1)] \begin{bmatrix} \xi_1'(\bar{\tau}_1) & \cdots & \xi_n'(\bar{\tau}_1) \\ \vdots & \ddots & \vdots \\ \xi_1'(\bar{\tau}_n) & \cdots & \xi_n'(\bar{\tau}_n) \end{bmatrix}^{-1}. \tag{2.30}$$

Since the quadrature weights $\mathbf{w}_\mathrm{q}$ only depend on $[\bar{\tau}_0, ..., \bar{\tau}_n]$, they only have to be computed once for a given n.

Using direct collocation for discretization as outlined above, (2.8) turns into

$$\mathbf{c}_{\mathrm{inv},0,0} = \left[\mathbf{g}(\mathbf{x}_{0,0}, \mathbf{u}_0), \dot{\mathbf{g}}(\mathbf{x}_{0,0}, \mathbf{u}_0)\right]^\mathsf{T} = \mathbf{0}, \tag{2.31}$$

where the coordinate transformation T_c is used again to map from $[\mathbf{q}, \dot{\mathbf{q}}, t]$ to $[\mathbf{x}, \mathbf{u}]$.

Note that $\mathbf{g}$ and $\dot{\mathbf{g}}$ can both explicitly depend on the control $\mathbf{u}$ depending on the relative degree of the dynamical system $\Sigma : \mathcal{U} \to \mathcal{X}, \mathbf{u}(\cdot) \mapsto \mathbf{x}(\cdot)$, where the notation $(\cdot)$ denotes the entire trajectory of a decision variable. Hence, Σ is defined as a map from an input trajectory to an output trajectory. If the relative degree of Σ is e.g. two, then $\ddot{\mathbf{g}}$ is a function of $\mathbf{u}$.

In this formulation, all constraints are discretized by only enforcing them for the OCP's decision variables, namely $\mathbf{x}$, $\mathbf{u}$ and $\boldsymbol{\lambda}$ on the control points. It is still possible that the constraints are violated between the control points.

2.3. Interior-Point Methods for Numerical Optimization

Now that (2.9) has been discretized using direct collocation, we need a way to numerically solve the discrete OCP (2.22). For this, we choose to use an IP method. To explain the basic working principles of IP methods, we restate (2.22), but without explicitly writing down the box constraints that are instead fused into the general inequality constraint

vector $\mathbf{h}$, whereas the collocation and continuity constraints are fused into the equality constraint vector $\mathbf{c}$. Also, we summarize all decision variables as $\mathbf{w} \in \mathbb{R}^{n_\mathbf{w}}$, with $n_\mathbf{w} = n_\mathbf{x} + n_\mathbf{u} + n_{\boldsymbol{\lambda}} + 1 + n_\mathbf{p}$.

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & J(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{0} \geq \mathbf{h}(\mathbf{w}), \\ & \mathbf{0} = \mathbf{c}(\mathbf{w}). \end{aligned} \tag{2.32}$$

For the inequality constraint vector $\mathbf{h} = [h_0, ..., h_{i_\mathbf{h}}, ..., h_{n_\mathbf{h}}] \in \mathbb{R}^{n_\mathbf{h}}$, an inequality constraint h_{i_h} is called active at a feasible point $\mathbf{w}^*$, if and only if $h_{i_h}(\mathbf{w}^*) = 0$. We call the set of all active inequality constraints the active set $\mathcal{A}_\mathbf{h}$. A point $\mathbf{w}^*$ is further said to fulfill the linear independence constraint qualification (LICQ), if the matrix

$$\mathbf{L} = \begin{bmatrix} \nabla_\mathbf{w} \mathbf{c}(\mathbf{w}^*) \\ \nabla_\mathbf{w} \mathbf{h}_a(\mathbf{w}^*) \end{bmatrix} \in \mathbb{R}^{(n_\mathbf{c} + n_{\mathbf{h}_a}) \times n_\mathbf{w}} \tag{2.33}$$

has full rank, where $\mathbf{h}_\mathrm{a} \in \mathbb{R}^{n_{\mathbf{h}_a}}$ is a concatenation of all $h_{i_\mathbf{h}} \in \mathcal{A}_\mathbf{h}$. This is equivalent to the condition that the gradients of all equality constraints and all active inequality constraints must be linearly independent.

Furthermore, we define the Lagrange function of (2.32) as

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = J(\mathbf{w}) + \boldsymbol{\lambda}^\mathsf{T} \mathbf{c}(\mathbf{w}) + \boldsymbol{\mu}^\mathsf{T} \mathbf{h}(\mathbf{w}), \tag{2.34}$$

with the Lagrange multiplier vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. Note that in the Lagrange function, equality and inequality constraints are treated separately with a separate vector of multipliers each.

The OCP (2.32) can be solved by looking for points that satisfy the so called Karush-Kuhn-Tucker (KKT) conditions [32]:

Theorem 1 (KKT conditions)**.** *If $\mathbf{w}^*$ is a local minimizer of the optimization problem defined in (2.32) and if LICQ holds at $\mathbf{w}^*$, then there exist multiplier vectors $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$ such that*

$$\nabla_\mathbf{w} \mathcal{L}(\mathbf{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \mathbf{0} \tag{2.35}$$

$$\mathbf{c}(\mathbf{w}^*) = \mathbf{0} \tag{2.36}$$

$$\mathbf{0} \geq \mathbf{h}(\mathbf{w}^*) \perp \boldsymbol{\mu}^* \geq \mathbf{0}. \tag{2.37}$$

The orthogonal operator $\perp$ implies that the vectors $\mathbf{h}$ and $\boldsymbol{\mu}$ are orthogonal to each other. This means that for every $h_{i_\mathbf{h}} \notin \mathcal{A}_\mathbf{h}$, $\mu_{i_\mathbf{h}} = 0$. In general, the KKT conditions are only a necessary condition for optimality. They are also a sufficient condition only for convex problems.

Since we usually deal with non-convex problems, we formulate a second-order sufficient condition [32]. To do so, we define $\mathcal{Z}$ as the null-space of the matrix

$$\mathbf{A} = \begin{bmatrix} \nabla_\mathbf{w} \mathbf{c}(\mathbf{w}^*) \\ \nabla_\mathbf{w} \mathbf{h}^+(\mathbf{w}^*) \end{bmatrix} \in \mathbb{R}^{n_\mathbf{A} \times n_\mathbf{w}}, \tag{2.38}$$

where $\mathbf{h}^+$ are the strictly active inequality constraints. A constraint $h_{i_\mathbf{h}}$ is strictly active if and only if $h_{i_\mathbf{h}} \in \mathcal{A}_\mathbf{h}$ and $\mu_{i_\mathbf{h}} > 0$. The matrix $\mathbf{Z} \in \mathbb{R}^{n_\mathbf{w} \times (n_\mathbf{w} - n_\mathbf{A})}$ is defined with the conditions that $\mathbf{AZ} = \mathbf{0}$, $\mathbf{Z}^\mathsf{T}\mathbf{Z} = \mathbf{0}$ and that the columns of $\mathbf{Z}$ are a base of $\mathcal{Z}$. The second-order condition can then be formulated as:

Theorem 2 (Strong second-order sufficient conditions for optimality)**.** *If* $(\mathbf{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ *fulfills the KKT conditions and the LICQ and if the Hessian of its Lagrangian is positive definite on the corresponding space* $\mathcal{Z}$*, i.e., if*

$$\mathbf{Z}^\mathsf{T} \nabla^2_\mathbf{w} \mathcal{L}(\mathbf{w}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{Z} > \mathbf{0}, \tag{2.39}$$

then the point $\mathbf{w}^*$ *is a local minimizer of the problem formulated in (2.32).*

For an OCP without inequality constraints, theorem 1 can be converted easily into a nonlinear root-finding problem that can then be solved by using some sort of Newton-type method. For OCPs with inequality conditions, the so called complementary slackness condition (2.37) makes this straightforward approach impossible. IP methods try to approximate condition (2.37) in such a way that fast and reliable Newton-type methods can be used to find points that fulfill the KKT conditions of theorem 1 and also ensure that the condition (2.39) from theorem 2 holds.

As a first step, we introduce the slack variables $\mathbf{s}$ to rewrite the constraint $h_{i_\mathbf{h}}(\mathbf{w}) \leq 0$ as $h_{i_\mathbf{h}}(\mathbf{w}) + s_{i_\mathbf{h}} = 0$ together with $s_{i_\mathbf{h}} > 0$. Secondly, we add a barrier function term to the cost function such that

$$\bar{J}(\mathbf{w}, \mathbf{s}, \tau) = J(\mathbf{w}) - \tau \sum_{i_\mathbf{h}=1}^{n_\mathbf{h}} \log s_{i_\mathbf{h}} \tag{2.40}$$

is the new cost function with the barrier parameter $\tau > 0$. One can verify that for $\tau \to 0$, the barrier function acts as an ideal indicator function for $s_{i_\mathbf{h}} > 0$, vanishing for any strictly positive $s_{i_\mathbf{h}}$ and going to infinity for $s_{i_\mathbf{h}} \to 0$. However, by decreasing τ, the numerical properties of the barrier function also deteriorate because it converges point-wise to the discontinuous indicator function.

Using the new cost function (2.40), we can formulate the OCP

$$\begin{aligned} \underset{\mathbf{w},\,\mathbf{s}}{\text{minimize}} \quad & \bar{J}(\mathbf{w}, \mathbf{s}, \tau) \\ \text{subject to} \quad & \mathbf{0} = \mathbf{h}(\mathbf{w}) + \mathbf{s}, \\ & \mathbf{0} = \mathbf{c}(\mathbf{w}), \end{aligned} \tag{2.41}$$

which is only constrained by equality constraints. Introducing the slack variable $\mathbf{s}$ and adding the barrier term to the cost function approximates the inequality constraints without actually formulating them in the OCP. We can now apply theorem 1 to (2.41), resulting in

$$\nabla_\mathbf{w} \bar{\mathcal{L}}(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \tau) = \mathbf{0} \tag{2.42}$$

$$\mathbf{c}(\mathbf{w}) = \mathbf{0} \tag{2.43}$$

$$\mathbf{h}(\mathbf{w}) + \mathbf{s} = \mathbf{0}, \tag{2.44}$$

to which we add

$$\mu_{i_\mathbf{h}} s_{i_\mathbf{h}} - \tau = 0, \qquad \text{for} \quad i_\mathbf{h} = 1, ..., n_\mathbf{h}. \tag{2.45}$$

Equation (2.45) is a smooth approximation of the complementary slackness condition for $s_{i_\mathbf{h}} > 0$, the inequality conditions that were not explicitly stated in (2.41). For $\tau \to 0$ the original complimentary slackness condition is recovered. Equations (2.42) – (2.45) constitute the desired root-finding problem that can be solved using Newton-type methods.

It has already been mentioned that $\tau \to 0$ gives the best approximation of (2.32), but results in bad numerical behavior because the cost function $\bar{J}$ would then be (almost) non-continuous. Hence, a homotopy is employed to solve (2.41). First, it is solved for a large τ_0, whereafter the solution is used as an initial guess for the same problem with a smaller τ_1. This procedure is repeated for a series of τ_{i_τ}, $i_\tau \in \{0, 1, ..., n_\tau\}$, with $\tau_{i_\tau} > \tau_{i+1} > 0$. The value of the last τ_{n_τ} is a measure of how accurately (2.41) approximates (2.32) with respect to the inequality constraints. The engineer using the IP method has to choose a value based on the classical trade-off between computation time and accuracy.

Due to the homotopy that is performed during the IP algorithm, one does not directly solve for an optimal value $\mathbf{w}^*$ but rather a series of $\mathbf{w}^*_{i_\tau}$ that are each optimal with regard to the barrier parameter τ_{i_τ} of the corresponding homotopy step. This means that $\mathbf{w}^*_{i_\tau+1}$ is a suboptimal solution to the i_τ-th homotopy step. Thus, it is difficult to warm start an IP method because even if the solver is initialized with the optimal solution $\mathbf{w}^*_{n_\tau}$, it first has to solve all intermediate problems with their corresponding τ_{i_τ}, resulting in a series of $\mathbf{w}^*_{i_\tau}$. It is even possible that the optimal solution $\mathbf{w}^*_{n_\tau}$ is infeasible for another τ_{i_τ} if $i_\tau \neq n_\tau$. Hence, it is a non-trivial question how to choose the series of τ_{i_τ} and especially how to choose τ_0 and τ_{n_τ}.

Since the constraints $\mu_{i_\mathbf{h}} > 0$ and $s_{i_\mathbf{h}} > 0$ are not explicitly stated in the root-finding problem (2.42) – (2.45), one has to shorten the step size during the Newton-type iterations to ensure that the constraints are met. One also has to ensure within the Newton-type method that the condition of theorem 2 holds by regularizing the Hessian in such a way that it is positive definite during each step of the Newton-type method.

Compared to SQP methods, IP methods have two important advantages. Firstly, they only need to approximately solve a root-finding problem per homotopy step, while SQP methods need to solve an entire quadratic program (QP) of the same size in each iteration. Especially since some SQP solvers use IP methods to solve these QPs, IP methods are often less computationally expensive than SQP methods.

Secondly, since we can choose the accuracy with which the inequality constraints are approximated with the barrier parameter τ, a large τ_0 (i.e. a bad initial approximation of the inequality constraints) can facilitate the process of finding a feasible initial condition. In other words, if a bad initial guess $\mathbf{w}_0$ turns out to be infeasible for a given τ_0, it is possible to increase τ_0 until $\mathbf{w}_0$ is feasible without necessarily decreasing the quality of the solution $\mathbf{w}^*_{n_\tau}$. This is especially useful for complicated and unintuitive mechanical

systems like multi-kite AWES for which the computation of feasible initial conditions is very difficult.

2.4. Homotopy Strategy

None of the problems discussed in this thesis can be robustly solved with a single OCP. Instead, we employ a homotopy strategy similar to [18, 24] and solve a series of OCPs $\mathcal{P}_l$, $l \in \{0, 1, ..., P\} = \mathcal{P}_\mathrm{H}$, one after the other, using the solution of OCP $\mathcal{P}_l$ as initial guess for $\mathcal{P}_{l+1}$. By doing so, we can first find a solution to a problem $\mathcal{P}_0$, for which it is easy to find a feasible solution, and then gradually deform the problem to $\mathcal{P}_P$, the problem we actually want to solve. In order to explain the homotopy strategy, we will first elaborate its working principle and then present a more precise formalization.

Working Principle

For the homotopy strategy, we introduce two different homotopy parameter vectors. The vector $\boldsymbol{\phi} = [\phi_0, ..., \phi_l, ..., \phi_P]^\mathsf{T} \in \mathbb{R}^{P+1}$ consists of scalar homotopy parameters ϕ_l that are bounded on the interval $[0, 1]$. The vector $\boldsymbol{\theta} = [\boldsymbol{\theta}_0^\mathsf{T}, ..., \boldsymbol{\theta}_l^\mathsf{T}, ..., \boldsymbol{\theta}_P^\mathsf{T}]^\mathsf{T}$ consists of unbounded homotopy parameters that are used to soften constraints by expanding the feasible set of the OCP. They can but do not have to be scalar. Together the homotopy parameters are added to the OCP as an optimization variable $\mathbf{p} = [\boldsymbol{\phi}^\mathsf{T}, \boldsymbol{\theta}^\mathsf{T}]^\mathsf{T}$ in accordance with the notation of section 2.2[2].

There exists a scalar bounded homotopy parameter ϕ_l for every homotopy step $\mathcal{P}_l$. It is used with a homotopy function with a linear embedding that can be part of the cost function, the constraints or the dynamic equations of the OCP formulation.

Let us assume we have the homotopy

$$\mathbf{S}_l(\mathbf{p}) = \phi_l \mathbf{A}(\boldsymbol{\theta}_l) + (1 - \phi_l)\mathbf{B} \tag{2.46}$$

somewhere within the OCP formulation, where $\phi_l \in [0, 1]$ is the bounded homotopy parameter associated with step $\mathcal{P}_l$.

Note that $\mathbf{A}(\boldsymbol{\theta}_l)$ can be a function of the unbounded parameter vector $\boldsymbol{\theta}_l$. A possible interpretation of the function $\mathbf{A}(\boldsymbol{\theta}_l)$ is that of an unbounded fictitious force that can be introduced into the dynamics. Since $\boldsymbol{\theta}_l$ is an optimization variable, it facilitates finding a feasible solution by altering the dynamics of the system in a way that expands the feasible set of the OCP. After finding a feasible solution with the help of $\boldsymbol{\theta}_l$, we use the homotopy to shrink the feasible set to correspond to the physical forces in the dynamics. To do so, we perform homotopy step $\mathcal{P}_l$.

Within $\mathcal{P}_l$, we perform the following two sub-steps to deform the problem formulation from the starting function $\mathbf{A}$ to the target function $\mathbf{B}$:

[2]Technically, the homotopy parameters are optimization variables. However, we continue to call them parameters to distinguish them from the rest of the optimization variables.

- Solve the OCP for $\boldsymbol{\phi}$, $\boldsymbol{\theta}$ bounded element-wise by

$$\boldsymbol{\theta}_r = \mathbf{0}, \forall r < l \tag{2.47}$$
$$\phi_r = 0, \forall r < l \tag{2.48}$$
$$\phi_l \in [0, 1] \tag{2.49}$$
$$\phi_r = 1, \forall r > l, \tag{2.50}$$

  with the initial guess $\phi_l = 1$ and add a strong penalty for ϕ_l in the cost function. This results in $\mathbf{S}(\mathbf{p}) = \mathbf{A}(\boldsymbol{\theta}_l)$ at the beginning of the homotopy step, whereas ϕ_l is then decreased due to its strong penalty until $\phi_l \approx 0$ and $\mathbf{S}(\mathbf{p}) \approx \mathbf{B}$. Since $\boldsymbol{\theta}_l$ is unbounded, we must penalize it to make sure that $\mathbf{A}(\boldsymbol{\theta}_l)$ does not grow with the same rate that ϕ_l decreases.

- Solve the OCP with the bounds

$$\phi_r = 0, \forall r \leq l \tag{2.51}$$
$$\phi_r = 1, \forall r > l \tag{2.52}$$
$$\boldsymbol{\theta}_r = \mathbf{0}, \forall r \leq l. \tag{2.53}$$

  This makes sure that $\mathbf{S}(\mathbf{p}) = \mathbf{B}$. Consequently, the deformation from $\mathbf{A}$ to $\mathbf{B}$ is complete.

The term $\mathbf{A}$ does not have to be a function of $\boldsymbol{\theta}_l$, but can also be e.g. a constraint or a cost function term. We do not allow $\mathbf{B}$ to be a function of $\boldsymbol{\theta}_l$.

Homotopy Formalization

Now that we have explained the basic working principle of the homotopy strategy, we set out to find a precise formalization of it. Our goal in doing so is to develop a concise notation for the description of homotopy schedules. For this, we identify three separate parts of the OCP formulation that can contain homotopies: the cost function $J \in \mathbb{R}$, the dynamics $\mathbf{D} \in \mathbb{R}^{n_\mathbf{D}}$ and the constraints $\mathbf{C} \in \mathbb{R}^{n_\mathbf{C}}$.

The cost function takes the form of

$$J(\mathbf{p}) = J_1(\boldsymbol{\phi}) + J_2(\boldsymbol{\phi}) + f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) + f_{\boldsymbol{\phi}}(\boldsymbol{\phi}) + J_0. \tag{2.54}$$

We will go over this notation bit by bit.

The first two terms are defined as

$$J_1(\boldsymbol{\phi}) = \sum_{l \in \mathcal{P}_\mathrm{H}} J_{1,l}(\phi_l) \tag{2.55}$$
$$J_2(\boldsymbol{\phi}) = \sum_{l \in \mathcal{P}_\mathrm{H}} J_{2,l}(\phi_l). \tag{2.56}$$

Here in turn we use the definition of

$$J_{1,l}(\phi_l) = \phi_l \tilde{J}_{1,l} = p^J(\tilde{J}_{1,l}) \tag{2.57}$$

$$J_{2,l}(\phi_l) = (1-\phi_l)\tilde{J}_{2,l} = p^J(\tilde{J}_{2,l}), \tag{2.58}$$

where we define the placement operator

$$p^{\mathbf{a}}(\mathbf{b}) = \mathbf{a}_{\mathbf{b}}\mathbf{b}, \tag{2.59}$$

for two vectors $\mathbf{a}$ and $\mathbf{b}$. We choose this notation to be able to simply state $\tilde{J}_{1,l}$ and $\tilde{J}_{2,l}$ for a given l, without having to further specify, where they appear in J. This will be especially useful when we later define the non-scalar expressions $\mathbf{D}$ and $\mathbf{C}$ in the same way.

The term J_0 denotes all terms of the cost function that are not functions of $\boldsymbol{\phi}$.

The functions $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ and $f_{\boldsymbol{\phi}}(\boldsymbol{\phi})$ are there to penalize the homotopy parameters $\mathbf{p}$ at the appropriate homotopy steps. To do so, we define the function $f_{\boldsymbol{\phi}}(\boldsymbol{\phi})$ as

$$f_{\boldsymbol{\phi}}(\boldsymbol{\phi}) = \sum_{l \in \mathcal{P}_{\mathrm{H}}} \tilde{r}_l \tilde{f}_{\phi_l}(\phi_l), \tag{2.60}$$

where $\tilde{f}_{\phi_l}$ are strictly quasi-convex functions with $\tilde{f}_{\phi_l}(0) = 0$ and $\tilde{r}_l$ is a scalar that is set according to

$$\tilde{r}_r = 0, \forall r < l \tag{2.61}$$

$$\tilde{r}_l = 1 \tag{2.62}$$

$$\tilde{r}_r = 0, \forall r > l. \tag{2.63}$$

This makes sure that ϕ_l is penalized according to $\tilde{f}_{\phi_l}(\phi_l)$ in the homotopy step $\mathcal{P}_l$. The function $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is defined analogously.

The dynamics and constraints are similarly expressed as

$$\mathbf{D}(\mathbf{p}) = \mathbf{D}_1(\mathbf{p}) + \mathbf{D}_2(\boldsymbol{\phi}) + \mathbf{D}_0 \tag{2.64}$$

and

$$\mathbf{C}(\mathbf{p}) = \mathbf{C}_1(\mathbf{p}) + \mathbf{C}_2(\boldsymbol{\phi}) + \mathbf{C}_0, \tag{2.65}$$

with the analogous definitions for $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_0$, $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_0$. Let us explicitly state the definitions (2.55) – (2.58) for the dynamics $\mathbf{D}$:

$$\mathbf{D}_1(\mathbf{p}) = \sum_{l \in \mathcal{P}_{\mathrm{H}}} \mathbf{D}_{1,l}(\phi_l, \boldsymbol{\theta}_l) \tag{2.66}$$

$$\mathbf{D}_2(\boldsymbol{\phi}) = \sum_{l \in \mathcal{P}_{\mathrm{H}}} \mathbf{D}_{2,l}(\phi_l), \tag{2.67}$$

with

$$\mathbf{D}_{1,l}(\phi_l, \boldsymbol{\theta}_l) = p^{\mathbf{D}}(\tilde{\mathbf{D}}_{1,l}) \tag{2.68}$$

$$\mathbf{D}_{2,l}(\phi_l) = p^{\mathbf{D}}(\tilde{\mathbf{D}}_{2,l}). \tag{2.69}$$

Here we can make two observations. Firstly, $\mathbf{D}_1$ is a function of $\mathbf{p}$, which also includes $\boldsymbol{\theta}$, as opposed to $J_1(\boldsymbol{\phi})$. We can easily extend our notation for J_1 to also be a function of $\boldsymbol{\theta}$, but since we use the unbounded homotopy parameter $\boldsymbol{\theta}$ to increase the feasible set of an OCP, it does not make sense for it to appear in a homotopy like (2.46) in the cost function. Secondly, defining $\mathbf{D}_{1,l}$ with the help of the placement operator enables us to define $\tilde{\mathbf{D}}_{1,l}$ without having to worry about the different vector dimensions of $\tilde{\mathbf{D}}_{1,l}$ and $\mathbf{D}$.

With this notation, we can describe a switch in the cost function as

$$\tilde{J}_{1,l} \xrightarrow[\boldsymbol{\theta}_l]{\phi_l} \tilde{J}_{2,l} \tag{2.70}$$

for some homotopy step $\mathcal{P}_l$. The notation of (2.70) is interpreted as follows: switch from $\tilde{J}_{1,l}$ to $\tilde{J}_{2,l}$ using the bounded homotopy parameter ϕ_l and the unbounded homotopy parameter $\boldsymbol{\theta}_l$. The cost on $\mathbf{p}$ is changed according to (2.61) – (2.63). The bounds on $\mathbf{p}$ are changed according to (2.47) – (2.53). A full homotopy step can then be described by

$$\mathcal{P}_l: \quad \tilde{J}_{1,l} \xrightarrow[\boldsymbol{\theta}_l]{\phi_l} \tilde{J}_{2,l} \tag{2.71}$$

$$\tilde{\mathbf{D}}_{1,l} \xrightarrow[\boldsymbol{\theta}_l]{\phi_l} \tilde{\mathbf{D}}_{2,l} \tag{2.72}$$

$$\tilde{\mathbf{C}}_{1,l} \xrightarrow[\boldsymbol{\theta}_l]{\phi_l} \tilde{\mathbf{C}}_{2,l}. \tag{2.73}$$

From now on, we concisely present (2.71) – (2.73) for a full series of homotopy steps $\mathcal{P}_l$, $l \in \mathcal{P}_{\mathrm{H}}$ in the form of table 2.1.

Table 2.1.: Generic homotopy schedule.

homotopy step	$\mathcal{P}_0$	$\cdots$	$\mathcal{P}_l$	$\cdots$	$\mathcal{P}_P$
objective	$\tilde{J}_{1,0} \xrightarrow[\boldsymbol{\theta}_0]{\phi_0} \tilde{J}_{2,0}$	$\cdots$	$\tilde{J}_{1,l} \xrightarrow[\boldsymbol{\theta}_l]{\phi_l} \tilde{J}_{2,l}$	$\cdots$	$\tilde{J}_{1,P} \xrightarrow[\boldsymbol{\theta}_P]{\phi_P} \tilde{J}_{2,P}$
dynamics	$\tilde{\mathbf{D}}_{1,0} \xrightarrow[\boldsymbol{\theta}_0]{\phi_0} \tilde{\mathbf{D}}_{2,0}$	$\cdots$	$\tilde{\mathbf{D}}_{1,l} \xrightarrow[\boldsymbol{\theta}_l]{\phi_l} \tilde{\mathbf{D}}_{2,l}$	$\cdots$	$\tilde{\mathbf{D}}_{1,P} \xrightarrow[\boldsymbol{\theta}_P]{\phi_P} \tilde{\mathbf{D}}_{2,P}$
constraints	$\tilde{\mathbf{C}}_{1,0} \xrightarrow[\boldsymbol{\theta}_0]{\phi_0} \tilde{\mathbf{C}}_{2,0}$	$\cdots$	$\tilde{\mathbf{C}}_{1,l} \xrightarrow[\boldsymbol{\theta}_l]{\phi_l} \tilde{\mathbf{C}}_{2,l}$	$\cdots$	$\tilde{\mathbf{C}}_{1,P} \xrightarrow[\boldsymbol{\theta}_P]{\phi_P} \tilde{\mathbf{C}}_{2,P}$

If one of the table's elements is marked with a grayed out $\times$, it means that the corresponding homotopy in J, $\mathbf{D}$ or $\mathbf{C}$ is zero. The unbounded homotopy parameter $\boldsymbol{\theta}_l$ is

omitted when the homotopy does not depend on $\boldsymbol{\theta}_l$. For any series of homotopy steps, we initialize with the bound $\boldsymbol{\phi} = \mathbf{1}$, meaning that homotopy (2.46) has a default value of $\mathbf{S}_l = \mathbf{A}$ until it is changed.

This procedure has two main advantages over choosing a fixed step for the homotopy parameters $\mathbf{p}$. For one, instead of having to solve an arbitrary number of OCPs, we only have to solve two. Also, we can leave the choice of the step size for the homotopy parameters up to the sophisticated line-search algorithms of the numerical backend. The downside of this procedure is however that leaving the choice of step size up to the numerical backend also means that we have to use caution when designing $f_{\boldsymbol{\phi}}(\boldsymbol{\phi})$. If $\boldsymbol{\phi}$ is weighted too aggressively in the cost function, the step size in $\boldsymbol{\phi}$ might become so large that the solver "jumps" over the homotopy path.

In order to solve the OCPs for the emergency landing procedures, one first solves a problem for which an initial guess can be found more easily and then gradually changes the cost function, dynamics and constraints until arriving at the actual problem one wants to solve. Since we not only solve one problem but a series of problems $\mathcal{P}_l$, most of the $\mathcal{P}_l$ are only intermediate solutions. For an intermediate problem $\mathcal{P}_l$, it does not make sense to introduce all of the problem's nonlinearity by solving it for a whole range of barrier parameters τ. A sensible way of handling the nonlinearities is to set the initial barrier parameter τ_0 and the target barrier parameter τ_f differently for $\mathcal{P}_0$, $\mathcal{P}_P$ and all other intermediate OCPs $\mathcal{P}_l$. Specifically, we define the values τ_a, τ_b and τ_c with $\tau_\mathrm{a} > \tau_\mathrm{b} > \tau_\mathrm{c} > 0$. We then set τ_0 and τ_f according to table 2.2. With this approach, step $\mathcal{P}_0$ is only solved up to a barrier parameter of τ_b. The following problems $\mathcal{P}_l$ stay at this barrier parameter and only the final OCP $\mathcal{P}_P$ is solved with a low barrier parameter τ_c. Additionally, we distinguish between two error tolerances r_a and r_b for the interior-point solver, where $r_\mathrm{a} > r_\mathrm{b}$. The numerical values for the parameters presented in table 2.2 can be looked up in appendix A.3. The initial and terminal homotopy step are only used to change the settings of the solver. Hence, the whole columns corresponding to $\mathcal{P}_0$ and $\mathcal{P}_{\mathcal{P}_P}$ are grayed out. These homotopy steps only consist of one sub-step instead of two and neither penalize $\boldsymbol{\theta}$ nor $\boldsymbol{\phi}$. This information is not included in the following homotopy schedules since it is redundant for all schedules of this thesis.

Table 2.2.: Barrier parameters for $\mathcal{P}_l$.

	τ_0	τ_f	r_tol
$\mathcal{P}_0$	τ_a	τ_b	r_a
$\mathcal{P}_l$	τ_b	τ_b	r_a
$\mathcal{P}_P$	τ_b	τ_c	r_b

This chapter served to introduce the theory and notation that will be used in later chapters. We discussed how to model physical systems in an OC context using DAEs and how to discretize an OCP using direct collocation. Furthermore, we explained the

working principle of interior-point methods and the homotopy strategy that will be used to solve the OCPs of this thesis. We aimed to develop a notation for discretized OCP formulations and homotopy schemes that is both precise and easy to read.

3. Emergency Response Concept

The upcoming chapters of this thesis are mostly concerned with formulating and solving OCPs, thereby generating optimal trajectories tailored for different scenarios. The current chapter shall serve as an overall context in which all of these different trajectories should be viewed. It explains the general concepts and strategies of how a multi-kite AWES should deal with emergency scenarios. Section 3.1 starts by categorizing all emergency scenarios that that have been identified over the course of this thesis and elaborates on each of these categories. Section 3.2 then explains how emergencies will be detected by and internally represented within the system using a state machine. Lastly, section 3.3 concerns itself with the structure of the emergency landing strategies that are used in response to the detected emergency scenarios.

In order to make the AWES more safe and reliable, most components are implemented in a redundant fashion. Especially the sensors of the AWES must be redundant so that the long or short term failure of a single sensor does not render the whole AWES "blind".

3.1. Expected Emergency Scenarios

Section 1.3 has already given a short introduction to possible emergency scenarios. The following gives a more complete and structured view of the topic. All emergency scenarios that have been identified can be categorized as follows:

- **Sensor malfunction:** One or more of the kites' sensors breaking down can result in problems with sensor fusion and state estimation. False or missing sensor values can result in the state estimation converging to a wrong value or not converging at all, rendering the control setup instable. Since the kites' sensors are redundant, the suggested course of action is to act before further malfunctions result in an impeded maneuverability of the kites that would make a nominal landing procedure impossible.

- **Actuator malfunction:** The kites are actuated by servo-motors that are susceptible to attrition and bad weather conditions. A malfunction of the actuators leads to limited control possibilities of the kite in question. More specifically, the actuators can get stuck in a certain position, only work within a limited range or break down completely. This is not only true for the actuator on the kites but also for the generator winch in the ground-station.

- **Structural damage:** Bird strike or other unforeseen circumstances can inflict structural damage on the AWES, ranging from a deformation of the kites to the breaking of a tether. This can lead to a change of the kites' aerodynamics.
- **Software errors:** Even after careful debugging, it is possible that the code running on the kites fails. For example, it is possible for embedded algorithms to fail, e.g. by not converging in time for the next control interval.
- **Energy system malfunction:** The on-board battery or power distribution system can fail. It is to be expected that the on-board battery loses maximum capacity over time. Also, bad weather conditions like too much moisture or previous failure of electronic components can trigger a failure of the energy distribution system.
- **Electronics malfunction:** Due to moisture, the on-board electronics can malfunction. An error in the electronics is likely to trigger failures in the other system components that are connected by the electronics.
- **Communication errors:** Both the on-board communication and the communication between kites and ground-station can fail.

Looking at the above list, it is obvious that not all of these emergency scenarios can be effectively resolved by the means of an OC approach using trajectory optimization. Nevertheless, all emergency scenarios must be considered to come up with a unified methodology of how to handle them.

Since every nominal or emergency landing makes the AWES less efficient, there is a strong incentive to perform as few landings as possible. However, a nominal landing is much safer compared to an emergency landing and so it is reasonable to land while the malfunctioning components redundancy still ensures a nominal flight behavior for the AWES. In a mechatronic system, the failure of one component is often an indicator for the failure of other components shortly after. If one redundant sensor fails due to low temperatures, it is very likely that other sensors will follow suit since they are subject to the same temperatures. In addition, there is the concept of cascading failures: A malfunction in one part of a mechatronic system is likely to trigger other parts of the system to malfunction. A common example for this is the lighting on a bike: If the bulb of the front light burns through, the bicycle's dynamo provides too much voltage to the back light, causing it to break soon after if the front bulb is not exchanged. Considering these two arguments, it is unwise to try to improve the AWES efficiency by tolerating too many failures of redundant system components before landing and it should only be attempted if there is strong evidence that the failure was indeed an isolated incidence.

3.2. Emergency Detection

The first step in each emergency scenario is the detection of the emergency such that the proper cause of action can be chosen. We choose to formalize this basic procedure as a

state machine as shown in Fig. 3.2. The state machine's nodes correspond to operational modes, while the edges correspond to actions performed within these operational modes. By performing the actions attributed to the edges, the system's state can switch from node to node in the direction assigned by the arrows. The state machine has four nodes that are defined as follows:

- **Nominal** (N): The system is running smoothly without any problems. This is the system's desired operational mode, in which the system will ideally spend the majority of its time.
- **Warning** (W): A problem has occurred that needs further investigation and should be kept under observation. The response to these problems is dependent on how frequently and continuously they appear. The explosion of a sensor's variance for example can be completely harmless if it is an isolated, short-term phenomenon due to noise but needs to be monitored to ensure that it does not happen continuously, indicating a broken sensor. As far as the AWES's control policy is concerned, the state W is equivalent to the nominal state N.
- **Error** (E): An error has occurred that necessitates an immediate emergency landing. As opposed to a problem that only warrants a warning, an error is issued under the assumption that the problem in question is not an isolated short-term phenomenon, cannot be fixed while the AWES is still airborne and renders the AWES unsafe to fly. This does not necessarily mean that the AWES's controls are already compromised. Ideally the error state E is entered before the AWES loses its nominal flight behavior, but when it can be predicted that the AWES is about to be compromised in the foreseeable future, e.g. due to the failure of a redundant sensor.
- **Critical** (C): A critical error has occurred that renders the control setup effectively useless. Once the critical state C has been entered, no guarantee can be given that the control setup will produce useful output, e.g. when an uncaught exception crashes the flight controller.

The edges of the state machine correspond to actions performed within the given states, namely the detection of different kinds of emergencies. Each time one of these actions is performed, the system moves along the arrow and enters a new state. The edges are assigned as follows:

- d_{W}: A warning is issued. This means that a problem that necessitates a warning is detected. Note that the edge d_{W} does not necessarily lead to the warning state W.
- d_{N}: A problem that resulted in a previous warning is resolved. The resolution of such a problem requires more than the non-performance of d_{W}. In other words, it is not enough that a warning is no longer issued. There needs to be a change in the system's behavior indicating that the problem has indeed been resolved.

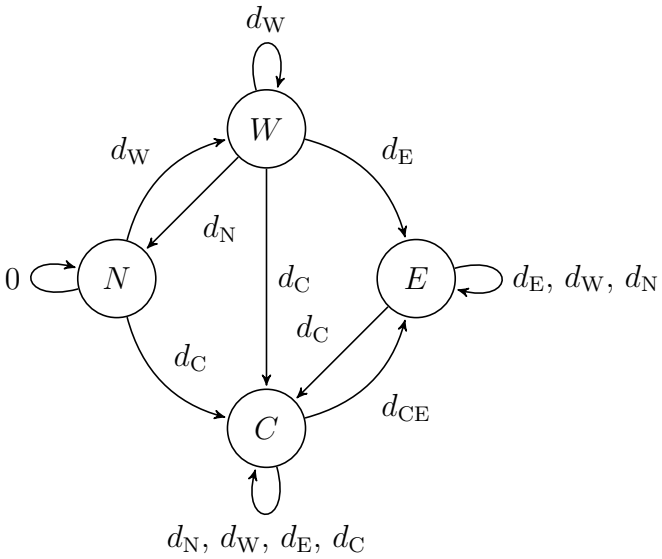

Figure 3.1.: State machine for error detection.

- d_E: An error occurred and has been detected by the system. As opposed to the issuing of a warning, all edges designated d_E lead to the error state E.
- d_C: A critical error occurred and has been detected by the system. Similar to d_E, all edges designated d_C lead to the critical state C.
- d_CE: A critical error has been resolved, e.g. by rebooting the flight controller. Since the AWES control setup is now functioning again, there is no reason not to use it to perform an appropriate emergency landing.

It is only ever possible to go along one edge at a time, meaning that e.g. a warning can only be issued if neither d_N nor d_E nor d_C are performed.

After a warning is issued and state W is entered, it is possible to go back along the edge d_N to enter the nominal state N. It is however not possible to return to state W or N once a (critical) error has been detected and state E or C has been entered. This design choice follows the idea that, once a (critical) error has been detected, the AWES should perform an emergency landing even if the problem resolves itself shortly after. This is due to the fact that such an event is less likely than that the evidence for said recovery is corrupted due to the preceding (critical) error. Hence, both edges d_N and d_W loop back from state E onto itself. The state can only be left by going along the edge d_C to state C. However, it is possible to go back from node C along the edge d_E and enter the state E again, in case that the functionality of the control setup is recovered in some way.

The choice which edge d to follow can be made by a Bayesian network using expert knowledge about the system [33]. The network must be set up in such a way that it

chooses the action d that corresponds to the most probable cause for a given set of prior assumptions. Hence, it performs a kind of artificial reasoning (AR). The choice of action must be unique for each set of prior assumptions to make sure that the system only ever moves along one edge at a time. Inside the states W, E and C there are other Bayesian networks set up to use AR to figure out which emergency scenario is most likely the cause of the prior assumptions that have been observed. This however will not be part of this thesis since it diverges too much from the topic of trajectory optimization. For the remainder of this thesis, we will therefore assume that the most likely emergency scenario has been deduced by a Bayesian network and is therefore known.

3.3. Hierarchical Emergency Landing Strategies

Section 3.2 has explained how emergency scenarios can be detected by the AWES and how a Bayesian network can be used to choose an appropriate response to a given scenario. However, we have not mentioned the possible responses so far. There are three different emergency landing strategies, (**A**), (**B**) and (**C**), ordered in descending desirability. Within each of these strategies, we assume that the emergency happens while the AWES is flying a pumping cycle, since this constitutes its most common operational mode. The logic as to when to apply each strategy is as follows:

Algorithm 1: Logic for choosing emergency landing strategy.

```
if (A) is possible then
    do (A);
    else if (B) is possible then
        do (B);
        else
            do (C);
        end
    end
end
```

Thus, strategy (**B**) is only employed if strategy (**A**) is not an option. Strategy (**C**) is only employed if both strategies (**A**) and (**B**) are not possible. Algorithm 1 will most likely not be implemented as code, since the choice of which strategy to use will be performed by this Bayesian network as outlined in section 3.2. The algorithm is displayed here to give a better understanding of the conceptual relationship of (**A**), (**B**) and (**C**). The three strategies are defined as follows:

- (**A**) **Transition to low altitude pumping trajectory**: Instead of landing, the AWES transitions into a low altitude pumping trajectory. This saves time and energy by omitting the inefficient landing and starting procedure. The strategy is applicable e.g. for when bad weather conditions make flying pumping cycles impossible but do not force the AWES to land. The AWES then flies the low

altitude pumping trajectory until the weather clears up again and then transitions back to flying pumping cycles. Since the AWES has nominal flight behavior in this case, there is no need to initiate the transition immediately. It is possible to continue on the pumping trajectory for a few seconds in order to find a more suitable starting point for the transition trajectory.

- (**B**) **Emergency landing with nominal flight behavior**: The AWES performs a landing in response to an emergency, but does so at a point where it still has nominal flight behavior. A likely scenario would be the failure of a redundant sensor or the on-board battery's maximum capacity getting too low. Similar to strategy (**A**), there is enough time to continue the pumping cycle until a desirable point is reached to initiate the landing trajectory. Since the AWES still has nominal flight behavior and the initial state can be freely chosen along the pumping trajectory, this strategy should always be successful, meaning that it should be able to land the AWES without damaging the environment or the system itself.
- (**C**) **Emergency landing with compromised flight behavior**: In case of a sudden emergency that compromises the AWES's flight behavior, there is no guarantee that the AWES can follow the pumping trajectory any further. Thus, the emergency landing trajectory needs to be initiated as soon as possible, no matter how inefficient the current position on the pumping cycle might be. This task is the most difficult to perform since the AWES is much more restricted in terms of its initial state and flight behavior. It is obvious that not all emergency scenarios permit an elegant landing trajectory that leave the AWES completely unharmed. In some cases, the optimal trajectory will not be the best, but rather the "least bad" resolution of the emergency scenario.

Both strategies (**B**) and (**C**) are designed as a two-phase landing procedure. In the first phase, the kites should fly close to the ground-station while reducing the AWES's kinetic energy using aerodynamic forces. In phase two, the kites' propellers are used to land the AWES. This two-phase landing procedure is put in place to prevent high relative velocities with regard to the wind from damage the AWES's propellers. In case of strategy (**C**) however, it can make sense to use the propellers right from the beginning of the trajectory, considering that a damaged propeller is favorable to other damages that could result from the AWES relying on aerodynamic forces for too long. However it must also be kept in mind that a propeller breaking in mid-flight is a great safety hazard due to its sudden influence on the AWES's maneuverability. Due to time constraints, this thesis will focus on the first part of the two-phase landing procedures.

This chapter served to introduce a set of comprehensive strategies for dealing with different kinds of emergency scenarios that are facing the multi-kite AWES. We will now go on to develop ways to mold these strategies into precise OCP formulations. In order to do so, it is first necessary to derive a model of the multi-kite AWES, which will be done in the next chapter.

4. DAE System Model

For feasibility studies and as a basis for future work, we use a point-mass model of the AWES. The derivation of the system's aerodynamics is taken from [45]. The Baumgarte stabilization and the choice of the state vector is similar to [17]. The point-mass model's obvious drawback is that it is only a 3 DOF model, lacking any explicit rotational dynamics. This lack of complexity however is also its biggest advantage, making it easier to intuitively understand how the system's states interact with each other. This makes the point-mass model ideal for a first implementation of the problems presented in this thesis. It is later possible to exchange the model for a more accurate 6 DOF formulation. The system is modeled using an implicit DAE $\mathbf{0} = \mathbf{f}$ just as explained in section 2.1.

The first section of this chapter gives an overview of the system's states, controls and coordinate system. The next section then goes on to discuss how the system's dynamics are derived with a Lagrange approach. Section 4.3 explains how wind and atmospheric properties are modeled. The next section goes into detail about the generalized forces that are acting on the system, specifically the aerodynamic kite forces and the tether drag. Section 4.5 elaborates on how the on-board battery of the kites is modeled.

The parameters used in this chapter are summarized in appendix A.1 together with their corresponding numerical values. The Ampyx PowerPlane (2nd gen.) as described in [25, 27] is chosen as a reference for the kite parameters.

4.1. System States, Controls and Coordinate System

The schematic layout of a dual-kite AWES is shown in Fig. 4.1. The systems nodes are numbered $p \in \{1, ..., N\} = \mathcal{N}$ while $\mathcal{A}$ denotes the set of all nodes with a kite attached. The tethers are numbered $k \in \{1, ..., N_\mathrm{T}\} = \mathcal{T}$. We further define the map $P : p \to P(p)$ that maps each node p to its parent $P(p)$. The parent $P(p)$ is connected to p by the tether $k = p$ by moving along the tether in the direction of the ground-station. The parent $P(1)$ is defined as node 0, even though $0 \notin \mathcal{N}$.

The system states of the AWES consist of

$$\mathbf{x} = \left[\mathbf{q}^\mathsf{T}, \dot{\mathbf{q}}^\mathsf{T}, \mathbf{C}_\mathrm{L}^\mathsf{T}, \mathbf{\Psi}^\mathsf{T}, l_\mathrm{T}, \dot{l}_\mathrm{T}\right]^\mathsf{T} \in \mathbb{R}^{n_\mathbf{x}}, \tag{4.1}$$

where $n_\mathbf{x} = 6N + 2N_\mathrm{K} + 2$ and $\mathbf{q}$, $\dot{\mathbf{q}} \in \mathbb{R}^{3N}$ are concatenations of the position and velocity vectors $\mathbf{q}_p$, $\dot{\mathbf{q}}_p$ for all nodes $p \in \mathcal{N}$. Similarly, $\mathbf{C}_\mathrm{L}$, $\mathbf{\Psi} \in \mathbb{R}^{N_\mathrm{K}}$ are concatenations of the lift coefficients $C_{\mathrm{L},a}$ and roll angles Ψ_a for all kite nodes $a \in \mathcal{A}$, where N_K is the cardinality of

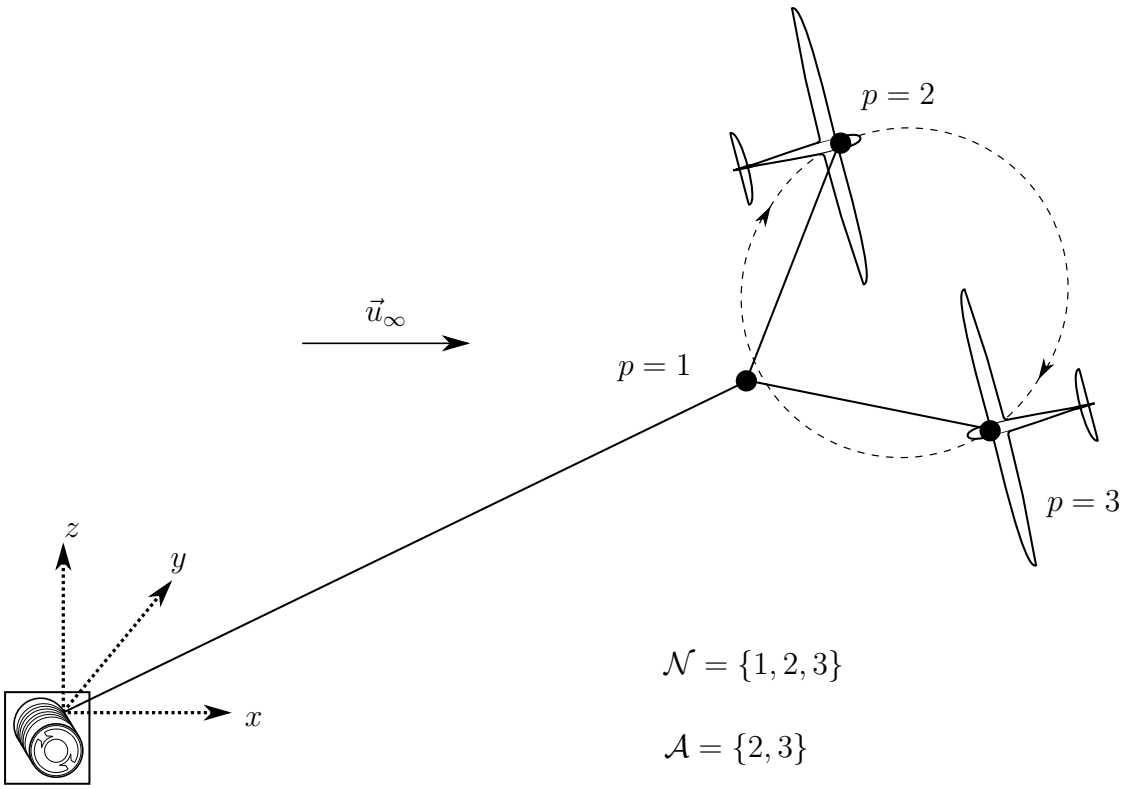

Figure 4.1.: Schematic sketch of the dual-kite AWES including the coordinate system and the labeling of the system's nodes [24].

$\mathcal{A}$. These expressions make up the state vector together with the global states, consisting of the main tether length and velocity l_T and $\dot{l}_\mathrm{T}$. The main tether velocity $\dot{l}_\mathrm{T}$ is defined in such a way that $\dot{l}_\mathrm{T} < 0$ means that the tether is reeled in towards the ground-station. For convenience, all states are summarized in table 4.1. It should be noted that since $\mathbf{q}_1^\mathsf{T}\mathbf{q}_1 - l_\mathrm{T}^2 = 0$, the chosen coordinates are not minimal. Even though the choice of coordinates is not minimal, one can show that modeling the system in these coordinates results in model equations of lesser complexity. Thereby the computational cost for evaluating the model equations and its corresponding sensitivities is reduced [45, 17]. Additionally, we have the generated energy E of the AWES, which is not included in the state vector $\mathbf{x}$ for ease of notation.

The controls of the system are chosen as

$$\mathbf{u} = \left[\dot{\mathbf{C}}_\mathrm{L}^\mathsf{T}, \dot{\mathbf{\Psi}}^\mathsf{T}, \ddot{l}_\mathrm{T}\right]^\mathsf{T} \in \mathbb{R}^{2N_\mathrm{K}+1}, \tag{4.2}$$

where $\dot{\mathbf{C}}_\mathrm{L}$, $\dot{\mathbf{\Psi}} \in \mathbb{R}^{N_\mathrm{K}}$ are defined analogously to $\mathbf{C}_\mathrm{L}$ and $\mathbf{\Psi}$. The system controls are the derivatives of the roll-angle Ψ and lift coefficient C_L of all kites as well as the acceleration of the main tether $\ddot{l}_\mathrm{T}$. This means that there are several undamped integrators in the

transfer function from the controls $\mathbf{u}$ to the states $\mathbf{x}$. The choice of $\dot{C}_\mathrm{L}$ and $\dot{\Psi}$ as controls implies that in the case of no control input, both roll-angle and lift coefficient are perfectly stabilized.

Table 4.1.: Model states.

state	unit	meaning
$\mathbf{q}_p$	m	position vector of node p
$\dot{\mathbf{q}}_p$	m/s	velocity vector of node p
$C_{\mathrm{L},a}$	$[-]$	lift coefficient of kite a
Ψ_a	deg	roll angle of kite a
l_T	m	main tether length
$\dot{l}_\mathrm{T}$	m	main tether velocity

We use a coordinate system with x pointing into the dominant wind direction, z upwards and y as an addition to the right-hand system. This means that the coordinate system is fixed as long as the wind model is constant in time t. The origin of the coordinate system is located at the ground-station.

4.2. Lagrangian Mechanics

The dynamics of the system shown in Fig. 4.1 are derived using a Lagrange approach. To do so, we employ the generalized coordinates $\mathbf{q} = [\mathbf{q}_1^\mathsf{T}, \mathbf{q}_2^\mathsf{T}, \mathbf{q}_3^\mathsf{T}]^\mathsf{T}$ corresponding to the nodes' positions. Note that the generalized coordinates $\mathbf{q}$, which are used to derive the Lagrange mechanics, are only a part of the system's states $\mathbf{x}$. The system must behave in such a way that the kites' positions are always consistent with the lengths of the main and secondary tethers l_T and l_s. This means that the system states are restricted to the manifold $\mathcal{M}$ defined by the invariant constraint

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} \mathbf{q}_1^\mathsf{T}\mathbf{q}_1 - l_\mathrm{T}^2 \\ (\mathbf{q}_2 - \mathbf{q}_1)^\mathsf{T}(\mathbf{q}_2 - \mathbf{q}_1) - l_\mathrm{s}^2 \\ (\mathbf{q}_3 - \mathbf{q}_1)^\mathsf{T}(\mathbf{q}_3 - \mathbf{q}_1) - l_\mathrm{s}^2 \end{bmatrix} = \mathbf{0}. \tag{4.3}$$

Considering (4.3), we can derive the Lagrange dynamics just as outlined in section 2.1. The systems potential energy can be computed as

$$V = \sum_{p=1}^{3} m_p g q_{z,p}, \tag{4.4}$$

where m_p is the mass of node p, consisting of half the weight of all tethers directly connected to p and the mass of the kite for $p \in \mathcal{A}$. Variable $q_{z,p}$ is the height of node p

and g the standard gravity. The system's kinetic energy consists of the kinetic energy of each node p as well as the generator's kinetic energy and can be formulated as

$$T = \sum_{p=1}^{3} \frac{1}{2} m_p \dot{\mathbf{q}}_p^{\mathsf{T}} \dot{\mathbf{q}}_p + \frac{1}{4} m_{\mathrm{G}} \dot{l}_T, \tag{4.5}$$

assuming the generator drum is a solid cylinder with mass m_{G}. The non-conservative generalized forces $\mathbf{F}$ of the system consist of aerodynamic forces acting on the kites and tether drag, as will later be discussed in section 4.4. The system's conservative forces (e.g. gravity) are accounted for by the potential energy term in the Lagrange function (2.1). By performing index reduction, (2.2) turns into

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{F}, \quad \ddot{\mathbf{g}} = \mathbf{0} \\ \mathbf{c}_{\mathrm{inv}}(0) = \mathbf{0}, \end{aligned} \tag{4.6}$$

where we again use the notation

$$\mathbf{c}_{\mathrm{inv}}(0) = \left[\mathbf{g}(\mathbf{q}(0))^{\mathsf{T}}, \dot{\mathbf{g}}(\mathbf{q}(0), \dot{\mathbf{q}}(0))^{\mathsf{T}}\right]^{\mathsf{T}}. \tag{4.7}$$

Using Baumgarte stabilization as outlined in section 2.1, the differential equation (DE) for the consistency constraint becomes

$$\ddot{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}(\mathbf{u})) = -2\kappa \dot{\mathbf{g}}(\mathbf{q}, \dot{\mathbf{q}}) - \kappa^2 \mathbf{g}(\mathbf{q}). \tag{4.8}$$

Note that $\ddot{\mathbf{g}}$ is a function of the control $\mathbf{u}$ since $\ddot{\mathbf{q}}$ is a function of $\mathbf{u}$.

The overall system dynamics are obtained by concatenating (4.6) with the trivial dynamics and the DE for the energy E. The system's algebraic variables are the Lagrange multipliers $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^{\mathsf{T}}$. Each Lagrange multiplier is linked to one entry of the invariant constraint $\mathbf{g}(\mathbf{q})$ and so to one specific tether. The expression $\lambda_k l_k = \tau_k$ is equal to the tension in the tether k. Hence, one can formulate the DE for the generated energy E as

$$\dot{E} = \lambda_1 l_{\mathrm{T}} \dot{l}_{\mathrm{T}}. \tag{4.9}$$

4.3. Wind and Atmospheric Model

The wind velocity is modeled using a logarithmic wind profile. The free stream velocity is defined as

$$\mathbf{u}_\infty(z) = u_{\mathrm{ref}} \frac{\log\left(\frac{z}{z_0}\right)}{\log\left(\frac{z_{\mathrm{ref}}}{z_0}\right)} \mathbf{e}_x, \tag{4.10}$$

where $\mathbf{e}_x$ is the unit vector in x-direction, z the height, z_0 the roughness length, z_{ref} the reference height and u_{ref} the reference wind speed at the reference height. Since the wind model does not depend on time t, the coordinate system is fixed.

The atmospheric properties air density ρ and temperature T are modeled after [3] according to the International Standard Atmosphere model. The temperature as a function of the elevation is modeled as

$$T(z) = T_{\text{ref}} - \Gamma z, \tag{4.11}$$

where T_{ref} is the reference temperature and Γ is the average lapse rate in the atmosphere. Using this expression, the air density is modeled as a function of temperature and elevation as

$$\rho(T, z) = \rho_{\text{ref}} \left(\frac{T(z)}{T_{\text{ref}}} \right)^{\left(\frac{g}{\Gamma R_{\text{air}}} - 1\right)}, \tag{4.12}$$

where ρ_{ref} is the reference air density, R_{air} the specific gas constant for air and g the standard gravity. Since temperature $T(z)$ does not appear anywhere else in the model equations and is only a function of z, we summarize these two equations as

$$\rho(z) = \rho_{\text{ref}} \left(\frac{T_{\text{ref}} - \Gamma z}{T_{\text{ref}}} \right)^{\left(\frac{g}{\Gamma R_{\text{air}}} - 1\right)}. \tag{4.13}$$

4.4. Generalized Forces Acting on the System

The generalized forces of the system are all non-conservative forces, which violate the conservation of energy within the system, meaning that they are responsible for a flux of energy into or out of the system. This distinguishes non-conservative forces from conservative forces like gravity, which conserve the energy within the system. When modeling the dual-kite AWES, the generalized non-conservative forces comprise the system's aerodynamic forces acting on the kites, namely lift and drag, as well as the drag forces acting on the tethers.

4.4.1. Aerodynamic Kite Forces

In order to model the generalized forces acting on the kite nodes $a \in \mathcal{A}$, we define the airspeed relative to the wind as

$$\mathbf{u}_{\text{A},a} = \mathbf{u}_\infty(z_a) - \dot{\mathbf{q}}_a. \tag{4.14}$$

Furthermore, we define

$$\hat{\mathbf{e}}_p = \frac{\mathbf{q}_p - \mathbf{q}_r}{\|\mathbf{q}_p - \mathbf{q}_r\|} \tag{4.15}$$

to be the unit vector pointing in the direction of the tether attached to node $p \in \mathcal{N}$ and connecting it to its parent $P(p) = r$. The transversal and lift axis of kite $a \in \mathcal{A}$ are then defined as

$$\mathbf{e}_{\text{T},a} = \frac{\mathbf{u}_{\text{A},a} \times \hat{\mathbf{e}}_a}{\|\mathbf{u}_{\text{A},a} \times \hat{\mathbf{e}}_a\|} \tag{4.16}$$

and

$$\mathbf{e}_{\mathrm{L},a} = \mathbf{e}_{\mathrm{T},a} \times \mathbf{u}_{\mathrm{A},a}. \tag{4.17}$$

The lift force acting on the kite can then be formalized as

$$\mathbf{F}_{\mathrm{L},a} = \frac{1}{2}\rho S C_{\mathrm{L},a}(\cos(\Psi_a)\mathbf{e}_{\mathrm{L},a}\|\mathbf{u}_{\mathrm{A},a}\| + \sin(\Psi_a)\mathbf{e}_{\mathrm{T},a}\|\mathbf{u}_{\mathrm{A},a}\|^2), \tag{4.18}$$

with the wing surface S. The drag force acting on the kites is given by

$$\mathbf{F}_{\mathrm{D},a} = \frac{1}{2}\rho S C_{\mathrm{D},a}\|\mathbf{u}_{\mathrm{A},a}\|\mathbf{u}_{\mathrm{A},a}, \tag{4.19}$$

with the drag coefficient

$$C_{\mathrm{D},a} = C_{\mathrm{D}}^0 + C_{\mathrm{D}}^I (C_{\mathrm{L},a})^2, \tag{4.20}$$

the wing drag coefficient C_{D}^0 and the induced drag coefficient C_{D}^I. A schematic sketch of the aerodynamic forces can be seen in Fig. 4.2.

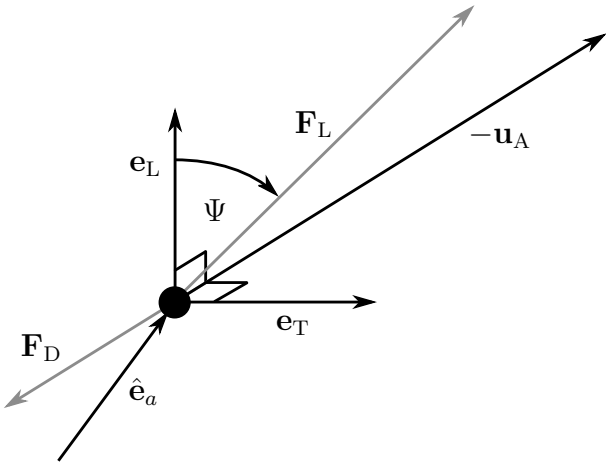

Figure 4.2.: Schematic sketch of the aerodynamic forces $\mathbf{F}_{\mathrm{L}}$ and $\mathbf{F}_{\mathrm{D}}$.

Within this 3 DOF model we assume that the relative velocity vector $\mathbf{u}_{\mathrm{A},a}$ of a kite a is exactly anti-parallel to its (implicit) orientation. Thus, neither side slip nor angle of attack appear in the model. Since the angle of attack does not appear in the model, stall is also not included. Hence, we have to assume that the bounds on $C_{\mathrm{L},a}$ are always set in a way that prevents the kite from stalling.

4.4.2. Tether Drag

The tethers are modeled as solid rods that cannot be subject to compressive forces. To model the tether drag forces for tether $k \in \mathcal{T}$, we define the tether's relative velocity

$$\mathbf{u}_{\mathrm{T},k} = \mathbf{u}_\infty(z_k) - \dot{\mathbf{q}}_{\mathrm{T},k}, \tag{4.21}$$

where $\dot{\mathbf{q}}_{\mathrm{T},k}$ and z_k are the average velocity and height of the two nodes attached to tether k.

Trivial Tether Drag Model

Following [45], the drag force acting on tether k is then expressed as

$$\mathbf{F}_{\mathrm{TD},k} = \frac{1}{2}\rho d_k l_k C_{\mathrm{TD},k} \|\mathbf{u}_{\mathrm{T},k}\| \mathbf{u}_{\mathrm{T},k} \tag{4.22}$$

and equally distributed to the nodes attached to tether k. This means that

$$\mathbf{F}^{\mathcal{N}}_{\mathrm{TD},p} = \sum_{k \in \mathcal{K}_\mathrm{a}} \frac{1}{2}\mathbf{F}_{\mathrm{TD},k}, \tag{4.23}$$

for all nodes $p \in \mathcal{N}$, where $\mathcal{K}_\mathrm{a}$ is the set of tethers that is attached to node p. The superindex $\mathcal{N}$ indicates that the index p is referring to a node, not a tether. The parameters d_k and l_k are diameter and length of tether k. The lift coefficient $C_{\mathrm{TD},k}$ is chosen to resemble that of a cylinder. This tether drag model will further be referred to as the trivial tether drag model.

Projected Tether Drag Model

A slightly more elaborate model can be derived by using the tether length that is perpendicular to the relative velocity of the tether instead of the whole tether length l_k. Equation (4.22) is thus transformed to

$$\mathbf{F}_{\mathrm{TD},k} = \frac{1}{2}\rho d_k l_{\perp,k} C_{\mathrm{TD},k} \|\mathbf{u}_{\mathrm{T},k}\| \mathbf{u}_{\mathrm{T},k}, \tag{4.24}$$

where $l_{\perp,k}$ is the tether length perpendicular to its own relative velocity. To derive $l_{\perp,k}$, we first derive the tether length parallel to the relative velocity as

$$l_{||,k} = \left(\frac{\mathbf{u}_{\mathrm{T},k}}{\|\mathbf{u}_{\mathrm{T},k}\|}\right)^{\mathsf{T}} (\mathbf{q}_k - \mathbf{q}_r), \tag{4.25}$$

where $\mathbf{q}_k$ and $\mathbf{q}_r$ are the positions of the two nodes attached to tether k. The perpendicular length is then derived by

$$l_{\perp,k} = \sqrt{l_k^2 - l_{||,k}^2}. \tag{4.26}$$

This tether drag model will further be referred to as projected tether drag model.

Equivalence Tether Drag Model

An even more sophisticated model of the tether drag can be derived using an equivalent forces approach. To do so, each tether $k \in \mathcal{T}$ is divided into n_T tether segments of identical length. The v-th segment of tether k is denoted as $s_{k,v}$ for $v \in \{1, ..., n_\mathrm{T}\}$. The

center of each segment $s_{k,v}$ is denoted as $\mathbf{q}_{\mathrm{T},k,v}$, the corresponding height and velocity as $\dot{\mathbf{q}}_{\mathrm{T},k,v}$ and $z_{k,v}$. The segments relative velocity is then

$$\mathbf{u}_{\mathrm{T},k,v} = \mathbf{u}_\infty(z_{k,v}) - \dot{\mathbf{q}}_{\mathrm{T},k,v}. \tag{4.27}$$

With this it is possible to compute the forces acting on $s_{k,v}$ by adapting the projected tether drag model (4.24) to

$$\mathbf{F}_{\mathrm{TD},k,v} = \frac{1}{2}\rho d_{k,v} l_{\perp,k,v} C_{\mathrm{TD},k,v} \|\mathbf{u}_{\mathrm{T},k,v}\| \mathbf{u}_{\mathrm{T},k,v}, \tag{4.28}$$

where $d_{k,v}$, $l_{\perp,k,v}$ and $C_{\mathrm{TD},k,v}$ are defined as above but for a segment $s_{k,v}$ and not a whole tether k. The forces and moments acting on tether k are now derived with

$$\mathbf{F}_{\mathrm{TD},k} = \sum_{v=1}^{n_\mathcal{T}} \mathbf{F}_{\mathrm{TD},k,v} \tag{4.29}$$

$$\mathbf{M}_{\mathrm{TD},k} = \sum_{v=1}^{n_\mathcal{T}} \mathbf{F}_{\mathrm{TD},k,v} \times \mathbf{l}_{\mathrm{M},k,v}, \tag{4.30}$$

where $\mathbf{l}_{\mathrm{M},k,v} = \mathbf{q}_{\mathrm{T},k} - \mathbf{q}_{\mathrm{T},k,v}$ is the corresponding moment arm. Here, $\mathbf{q}_{\mathrm{T},k}$ is the center of tether k. The forces $\mathbf{F}_{\mathrm{TD},k}$ and moments $\mathbf{M}_{\mathrm{TD},k}$ are now transformed into the body frame of tether k to yield $\mathbf{F}^{\mathrm{B}}_{\mathrm{TD},k}$ and $\mathbf{M}^{\mathrm{B}}_{\mathrm{TD},k}$. The z-axis in the body frame is pointing along the tether. The equivalent forces acting on the two endpoints of tether k are denoted as $\mathbf{F}^{\mathrm{B}}_{\mathrm{eq},k,\mathrm{l}}$ for the lower end and $\mathbf{F}^{\mathrm{B}}_{\mathrm{eq},k,\mathrm{u}}$ for the upper end. The components of the equivalent forces are

$$\mathbf{F}^{\mathrm{B}}_{\mathrm{eq},k,\mathrm{u}} = [f_{k,\mathrm{u},x}, f_{k,\mathrm{u},y}, f_{k,\mathrm{u},z}]^\mathsf{T} \tag{4.31}$$

$$\mathbf{F}^{\mathrm{B}}_{\mathrm{eq},k,\mathrm{l}} = [f_{k,\mathrm{l},x}, f_{k,\mathrm{l},y}, f_{k,\mathrm{l},z}]^\mathsf{T}. \tag{4.32}$$

The components of the forces and moments acting on the tether are

$$\mathbf{F}^{\mathrm{B}}_{\mathrm{TD},k} = [f_{k,x}, f_{k,y}, f_{k,z}]^\mathsf{T} \tag{4.33}$$

$$\mathbf{M}^{\mathrm{B}}_{\mathrm{TD},k} = [m_{k,x}, m_{k,y}, 0]^\mathsf{T}, \tag{4.34}$$

where we assume that tether k cannot rotate around its z-axis. With this notation, we can derive the equivalent forces by solving the linear system of equations (LSE)

$$\begin{bmatrix} f_{k,x} \\ f_{k,y} \\ f_{k,z} \\ m_{k,x} \\ m_{k,y} \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & \frac{l_k}{2} & 0 & 0 & -\frac{l_k}{2} & 0 \\ \frac{l_k}{2} & 0 & 0 & -\frac{l_k}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}}_{\mathbf{A}_{\mathrm{eq}}} \begin{bmatrix} f_{k,\mathrm{u},x} \\ f_{k,\mathrm{u},y} \\ f_{k,\mathrm{u},z} \\ f_{k,\mathrm{l},x} \\ f_{k,\mathrm{l},y} \\ f_{k,\mathrm{l},z} \end{bmatrix}. \tag{4.35}$$

Here the tether is assumed to be a line in $\mathbb{R}^3$ with no diameter in the body frame's x-y-plane. Thus, we enforce

$$0 = f_{k,\mathrm{u},z} - f_{k,\mathrm{l},z}. \tag{4.36}$$

The equivalent forces $\mathbf{F}^{\mathrm{B}}_{\mathrm{eq},k,\mathrm{u}}$ and $\mathbf{F}^{\mathrm{B}}_{\mathrm{eq},k,\mathrm{l}}$ can now be derived by inverting the matrix $\mathbf{A}_{\mathrm{eq}}$, which is always possible except for $l_k = 0$, which is not important since the length of tether k will always be strictly positive. We arrive at the desired forces $\mathbf{F}_{\mathrm{eq},k,\mathrm{u}}$ and $\mathbf{F}_{\mathrm{eq},k,\mathrm{l}}$ acting on the upper and lower end of tether k by performing the inverse transformation as before, this time from the body frame to the earth frame.

4.5. On-board Battery Model

The hardware on each kite is powered by an on-board battery that is charged by a propeller during flight. Since a compromised on-board battery is a potential cause for an emergency, it makes sense to include it in the model. We denote the energy stored in the on-board battery as E_{B}. We can then formulate

$$E_{\mathrm{B}}(t) = E_{\mathrm{B}}(t_0) + \int_{t_0}^{t} \dot{E}_{\mathrm{B}}(\bar{t})\ \mathrm{d}\bar{t}, \tag{4.37}$$

where the rate of change $\dot{E}_{\mathrm{B}}$ is defined as

$$\dot{E}_{\mathrm{B}} = \dot{E}_{\mathrm{B,in}} - \dot{E}_{\mathrm{B,out}}, \tag{4.38}$$

with $\dot{E}_{\mathrm{B,in}}$ denoting power going into the battery by charging of the battery during flight and $\dot{E}_{\mathrm{B,out}}$ denoting the power needed for the on-board hardware. Since we concern ourselves only with the battery during emergency scenarios, $\dot{E}_{\mathrm{B,in}}$ is not modeled in more detail and is set to zero anyways, thereby modeling a compromised on-board generator. The power needed for the on-board hardware is further divided into

$$\dot{E}_{\mathrm{B,out}} = \dot{E}_{\mathrm{B,con}} + \dot{E}_{\mathrm{B,el}} + \dot{E}_{\mathrm{B,act}}, \tag{4.39}$$

where the constants $\dot{E}_{\mathrm{B,con}}$ and $\dot{E}_{\mathrm{B,el}}$ characterize the upkeep needed to run the flight controller and the rest of the kite's electronics. The power needed for the actuation of the kite can be modeled as

$$\dot{E}_{\mathrm{B,act}} = M_{\mathrm{L}}\dot{\delta}_{\mathrm{L}} + M_{\phi}\dot{\delta}_{\Psi}, \tag{4.40}$$

where δ_{L} and δ_{Ψ} are the deflection angles of the control surfaces used to actuate the pitch and roll control of the kite. M_{L} and M_{Φ} are the moments acting on the servo-motors at a certain δ_{L}, δ_{Ψ}. The moment M_{L} is modeled as

$$M_{\mathrm{L}} = \bar{q} S_{\mathrm{L}} \frac{l_{\mathrm{L}}}{2} \sin(\delta_{\mathrm{L}}), \tag{4.41}$$

with the actuator length l_L, the actuator width w_L, the actuator surface $S_\mathrm{L} = l_\mathrm{L} w_\mathrm{L}$ and the dynamic pressure $\bar{q}$. The dynamic pressure $\bar{q}$ is

$$\bar{q} = \frac{1}{2}\rho v_\mathrm{rel}^2, \tag{4.42}$$

where v_rel is the control surface's relative velocity and can be expressed as $\|\mathbf{u}_{\mathrm{A},a}\|$. The dynamic pressure is multiplied with the effective control surface $S_\mathrm{L}\sin(\delta_\mathrm{L})$, meaning the part of the actuator surface that is orthogonal to $\mathbf{u}_{\mathrm{A},a}$, to yield the force acting on the control surface. The force is multiplied with the moment arm $\frac{l_\mathrm{L}}{2}$ to yield the moment acting on the servo-motor of the control surface. The moment arm is assumed to be half the length of the control surface. The moment M_Ψ can be modeled analogously as

$$M_\Psi = \bar{q} S_\Psi \frac{l_\Psi}{2}\sin(\delta_\Psi). \tag{4.43}$$

So far, the expression for $\dot{E}_\mathrm{B,act}$ is a function of δ_L, δ_Ψ and their corresponding derivatives with respect to time. Since the 3 DOF point-mass model only considers C_L and Ψ, we need to find expressions for $\delta_\mathrm{L}(C_\mathrm{L})$ and $\delta_\Psi(\Psi)$. To do so, we use Bryan's method according to [6, 31], which says that we can assume these expressions to take the form of

$$\delta_\mathrm{L}(C_\mathrm{L}) = \delta_{\mathrm{L},0} + C_{\delta,\mathrm{L}} C_\mathrm{L} \tag{4.44}$$
$$\delta_\Psi(\Psi) = \delta_{\Psi,0} + C_{\delta,\Psi}\Psi, \tag{4.45}$$

where $\delta_{\mathrm{L},0}$, $\delta_{\Psi,0}$, $C_{\delta,\mathrm{L}}$ and $C_{\delta,\Psi}$ are constants. We can estimate these constants by solving the LSE

$$\delta_{\mathrm{L},\max} = \delta_{\mathrm{L},0} + C_{\delta,\mathrm{L}} C_{\mathrm{L},\min} \tag{4.46}$$
$$\delta_{\mathrm{L},\min} = \delta_{\mathrm{L},0} + C_{\delta,\mathrm{L}} C_{\mathrm{L},\max} \tag{4.47}$$
$$\delta_{\Psi,\max} = \delta_{\Psi,0} + C_{\delta,\Psi}\Psi_\min \tag{4.48}$$
$$\delta_{\Psi,\min} = \delta_{\Psi,0} + C_{\delta,\Psi}\Psi_\max, \tag{4.49}$$

with given minimum and maximum deflections of the control surfaces. By plugging (4.44) – (4.45) into (4.40), the desired model is derived.

The energy stored in the on-board battery at the initial time $E_\mathrm{B}(t_0)$ can be expressed as a fraction of the energy stored on a fully charged battery. We compute the initial electrical energy stored on the battery as

$$E_{\mathrm{B},0} = N_\mathrm{B}\nu_\mathrm{fr}\nu_\mathrm{c} Q_\mathrm{B} U_\mathrm{B}, \tag{4.50}$$

where N_B is the number of battery cells, Q_B the charge, U_B the voltage, ν_c the conversion efficiency and ν_fr the fraction of how fully the battery is charged at the initial time.

In this chapter we have derived a 3 DOF model for a dual-kite AWES. This includes the aerodynamic kite forces as well as three different tether drag models of increasing complexity. We have also derived a model for the on-board battery of the kites.

The implicit DAE

$$\mathbf{0} = \mathbf{f}(\mathbf{x}, E, \dot{\mathbf{x}}, \dot{E}, \mathbf{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{n_{\mathbf{x}}+1+N_{\mathrm{T}}}, \tag{4.51}$$

which describes the model, consists of the DE for the states $\mathbf{x} \in \mathbb{R}^{n_{\mathbf{x}}}$, the DE for the energy $E \in \mathbb{R}$ and the (derivatives of the) algebraic equations $\ddot{\mathbf{g}} \in \mathbb{R}^{N_{\mathrm{T}}}$. In the case that the on-board battery energy and its derivative is added to the model as states for all kites, the dimension of $\mathbf{f}$ increases by $2N_{\mathrm{K}}$. For a dual-kite AWES without on-board batteries, there are 25 states (including the energy E), 5 controls, 3 algebraic equations and $\mathbf{f} \in \mathbb{R}^{28}$.

Now that we have obtained a model for the dual-kite AWES, we can continue with the formulation of a modular framework for the construction of OCPs that will be used to find optimal emergency trajectories.

5. Optimal Control Problem Formulation

All OCPs in this thesis are constructed out of modular building blocks to provide a framework that can be easily edited and precisely formalized. The current chapter gives an overview over all of these building blocks and how they are used. The building blocks are formulated in a discrete fashion such that they can be used in an OCP that has been discretized using direct collocation.

The first section of this chapter explains the different components that appear in the cost functions. The next section lists all the equality and inequality constraints that appear in the OCPs. Section 5.3 elaborates on the different homotopy steps that can be combined to a full homotopy scheme. Section 5.4 then goes into detail about how the initial guess for the initial homotopy step is constructed. The last section summarizes a number of numerical issues that have come up during the implementation of the OCPs.

After all of these concepts are introduced, chapters 7 and 8 will go into detail about each individual OCP and which building blocks they are made up of. A table containing all the numerical parameters and their corresponding values can be found in appendix A.3.

5.1. Cost Function

The cost functions of all the problems discussed in this thesis are constructed out of a number of modular cost components. More specifically, these components are called general problem cost G, tracking cost T, landing cost L and transition cost K. These components are then linked using the homotopy parameters $\boldsymbol{\phi} = [\gamma, \eta, \nu, \upsilon, \tau]^\mathsf{T}$ and $\boldsymbol{\theta}$, as is described in section 5.3.

General Problem Cost

The general problem cost comprises the regularization cost R and the homotopy parameter cost H. The regularization cost is defined as

$$R = \hat{R}_{\mathbf{u}} R_{\mathbf{u}} + \hat{R}_t R_t + \hat{R}_{\ddot{\mathbf{q}}} R_{\ddot{\mathbf{q}}}, \tag{5.1}$$

where $R_{\mathbf{u}}$, R_t, $R_{\ddot{\mathbf{q}}}$ are the regularization terms for the control input $\mathbf{u}$, the final time t_f and the node accelerations $\ddot{\mathbf{q}}$ respectively. The positive constants $\hat{R}_{\mathbf{u}}$, $\hat{R}_t$ and $\hat{R}_{\ddot{\mathbf{q}}}$ are

tuning parameters. The regularization cost R is added to the cost function to increase numerical stability and convergence.

The homotopy parameter cost H is defined as

$$H = H_\gamma + H_\eta + H_\nu + H_\upsilon + H_\tau + H_{\boldsymbol{\theta}}, \tag{5.2}$$

where

$$H_\gamma = \hat{H}_\gamma \gamma \tag{5.3}$$
$$H_\eta = \hat{H}_\eta \eta \tag{5.4}$$
$$H_\nu = \hat{H}_\nu \nu \tag{5.5}$$
$$H_\upsilon = \hat{H}_\upsilon \upsilon \tag{5.6}$$
$$H_\tau = \hat{H}_\tau \tau \tag{5.7}$$
$$H_\theta = \hat{H}_{\boldsymbol{\theta}} \boldsymbol{\theta}^\mathsf{T} \boldsymbol{\theta}, \tag{5.8}$$

with positive scalars $\hat{H}_\gamma$, $\hat{H}_\eta$, $\hat{H}_\nu$, $\hat{H}_\upsilon$, $\hat{H}_\tau$ and $\hat{H}_{\boldsymbol{\theta}}$. These tuning parameters differ depending on the homotopy step and are used to drive the corresponding homotopy parameter to zero as explained in section 2.4. The general problem cost is then expressed as

$$G = R + H. \tag{5.9}$$

Tracking Cost

The next cost component is the tracking cost T. It is defined as

$$T = \frac{1}{m} \left(\sum_{i=0}^{\bar{m}} \sum_{j=0}^{n} w_{\mathrm{q},j} (\mathbf{x}_{i,j} - \mathbf{x}_{\mathrm{ref},i,j})^\mathsf{T} \hat{\mathbf{T}} (\mathbf{x}_{i,j} - \mathbf{x}_{\mathrm{ref},i,j}) \right), \tag{5.10}$$

where $\hat{\mathbf{T}}$ is a positive definite square matrix and $\mathbf{x}_{\mathrm{ref}}$ is the reference state trajectory. The tracking cost T is quadratic with a minimum at $\mathbf{x} = \mathbf{x}_{\mathrm{ref}}$ and is thus used to drive $\mathbf{x}$ towards $\mathbf{x}_{\mathrm{ref}}$. The weights $w_{\mathrm{q},j}$ are defined as in (2.30) to ensure that the integral is approximated correctly using Radau quadrature. When using Radau quadrature, the sum is normalized with the fraction $\frac{t_\mathrm{f}}{m}$. Since we do not want the tracking cost T to linearly grow with the final time t_f, we only normalize with $\frac{1}{m}$.

Landing Cost

The landing cost L is used to generate trajectories with little final kinetic energy. It is defined as

$$L = \frac{1}{N} \left(\sum_{p=1}^{N} \dot{\mathbf{q}}_{p,\bar{m},n}^\mathsf{T} \hat{\mathbf{L}} \dot{\mathbf{q}}_{p,\bar{m},n} \right), \tag{5.11}$$

with the positive definite square matrix $\hat{\mathbf{L}}$. The landing cost penalizes the quadratic terminal velocity of every node.

Transition Cost

When the AWES is transitioning from one pumping cycle to another, the transition cost K is introduced. It is similar to the regularization cost and intended to favor transition trajectories with small node accelerations and little control input. The transition cost is defined as

$$K = \hat{K}_{\ddot{\mathbf{q}}} R_{\ddot{\mathbf{q}}} + \hat{K}_{\mathbf{u}} R_{\mathbf{u}}, \tag{5.12}$$

where $R_{\ddot{\mathbf{q}}}$ and $R_{\mathbf{u}}$ are the regularization terms of the node accelerations and controls respectively. The positive scalars $\hat{K}_{\ddot{\mathbf{q}}}$ and $\hat{K}_{\mathbf{u}}$ are tuning parameters.

5.2. Constraints

The constraints of the OCP are categorized by equality and inequality constraints. The equality constraints are discussed first, followed by the inequality constraints, which also include the variable bounds.

5.2.1. Equality Constraints

The OCP's equality constraints are all constraints that can be formulated as

$$\mathbf{0} = \mathbf{c}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \mathbf{p}) \tag{5.13}$$

in accordance with OCP (2.22). The most important equality constraints of the OCP are the collocation and continuity constraints, which constrain the optimal trajectory to follow the DAE outlined in chapter 4. Furthermore, they comprise an initial energy constraint and parameterized initial and terminal constraints.

Collocation and Continuity Constraints

Both collocation and continuity constraints have already been discussed at length in section 2.2. When adapted to the notation of the model derived in chapter 4, they are stated as

$$\mathbf{0} = \mathbf{c}_{\text{col}}(\mathbf{x}_{i,j}, E_{i,j}, \boldsymbol{\lambda}_{i,j}, \mathbf{u}_i, t_{\text{f}}, \mathbf{p}) \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}_- \tag{5.14}$$

$$\mathbf{0} = \mathbf{c}_{\text{con}}(\mathbf{x}_{i,n}, E_{i,n}, \mathbf{x}_{i+1,0}, E_{i+1,0}, \mathbf{p}) \qquad \forall i \in \mathcal{I}^-. \tag{5.15}$$

Enforcing the collocation constraints ensures that the DAE of the model holds at all collocation nodes. Enforcing the continuity constraints ensures that the states are continuous at the border of one collocation interval to the next.

Initial Energy Constraint

The generated energy E of the system is a relative quantity. It only denotes how much energy is generated or consumed by the system over the time interval $[0, t_\mathrm{f}]$, but does not specify the baseline to which the quantity is relative to. Since this baseline can be chosen arbitrarily, we set it to zero by enforcing

$$0 = E_{0,0}. \tag{5.16}$$

Parameterized Initial and Terminal Constraints

For emergency response strategy (**A**), we need to make sure that the optimal trajectories start and end on the given pumping trajectories. For strategies (**B**) and (**C**), one only ensures that the optimal trajectory starts on a given pumping trajectory. To formulate these constraints, we parameterize the given trajectories using `CasADi`-splines[1] as functions of a single normalized time parameter $\xi \in [0, 1]$. The parameterized trajectories are then denoted as $\hat{\mathbf{x}}_0(\xi_0)$ for the initial trajectory and $\hat{\mathbf{x}}_\mathrm{f}(\xi_\mathrm{f})$ for the terminal trajectory. With this notation, the two constraints

$$\mathbf{0} = \hat{\mathbf{x}}_0(\xi_0) - \mathbf{x}_{0,0} \tag{5.17}$$

$$\mathbf{0} = \hat{\mathbf{x}}_\mathrm{f}(\xi_\mathrm{f}) - \mathbf{x}_{\bar{m},n} \tag{5.18}$$

are formulated. Note that this notation is consistent with the exclusion of the energy E from the state vector $\mathbf{x}$ since we already enforce the initial energy to be zero. Furthermore, it does not make sense to assign the terminal energy to a specific value when transitioning from one pumping cycle to another. Also note that under the assumption that $\hat{\mathbf{x}}_0(\xi_0)$ is consistent for all $\xi_0 \in [0, 1]$, there is no need for an explicit consistency condition $\mathbf{c}_{\mathrm{inv},0,0} = \mathbf{0}$ to be introduced as a constraint. Depending on the specific problem we want to solve, the parameters ξ_0 and ξ_f are either introduced as optimization variables and bounded by the interval $[0, 1]$ or set to a specific value to ensure that the optimal trajectory starts on a specific point on the parameterized trajectory.

By enforcing $\xi \in [0, 1]$ we make sure that every point on the parameterized trajectory can be reached with the corresponding ξ. It is however not possible to go from $\xi = 0$ to $\xi = 1$ without passing through the whole interval during the optimization. Since the parameterized trajectories are usually pumping cycles and therefore periodic, it must be possible to pass from one end of the interval $[0, 1]$ to the other, since the points are identical. In practice however, this is not relevant since the parameterized initial constraint is very non-convex. Hence, we can assume that the constraint $\xi \in [0, 1]$ will never be active unless the initial guess for ξ is very close to the interval boundaries. This can be prevented easily by shifting the interval in such a way that the initial guess for ξ is approximately in the middle of the interval.

[1] `CasADi`-splines are a fast implementation of B-splines.

5.2.2. Inequality Constraints

The OCP's inequality constraints are all constraints that can be formulated as

$$\mathbf{0} \leq \mathbf{h}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \mathbf{p}) \tag{5.19}$$

in accordance with OCP (2.22). They comprise a tether stress constraint, an anti-collision constraint, a maximum acceleration constraint, a terminal position constraint, a minimum battery energy constraint as well as the state and control bounds. For ease of notation, we omit the indices i and j for the path inequalities. Unless otherwise specified, the following inequalities are defined for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$ (respectively $j \in \mathcal{J}_-$ in case of the algebraic variables).

Tether Stress Constraint

Since the tethers are modeled as permanently tight, one needs to enforce constraints to ensure that this assumption is valid. The tether stress constraint makes sure that the tether tension τ_k of all tethers $k \in \mathcal{T}$ is positive throughout the trajectory, thereby keeping the tethers tight. This can be formulated as

$$0 \leq \tau_k = l_k \lambda_k \tag{5.20}$$

for all tethers $k \in \mathcal{T}$. Since l_k is a length and therefore by definition positive, this is equivalent to enforcing that all λ_k are positive which can be expressed as

$$\mathbf{0} \leq \boldsymbol{\lambda}. \tag{5.21}$$

In addition to this, we also have to consider the tethers' material properties with respect to the maximum stress they can take. The maximum allowed stress is defined as

$$\sigma_{\max} = \frac{\sigma_{\mathrm{mat}}}{c_{\mathrm{s}}}, \tag{5.22}$$

where σ_{mat} is the maximum stress the tethers material can withstand and c_{s} is a safety factor. The maximum tether stress constraint is then expressed as

$$0 \leq \sigma_{\max,k} - \sigma_k = \sigma_{\max,k} - \lambda_k \frac{l_k}{A_k} \tag{5.23}$$

for all tethers $k \in \mathcal{T}$, where A_k is the cross section of tether k. To ensure that the constraint is scaled well, we reformulate it as

$$0 \leq \sigma_{\max,k} - \frac{\tau_k}{A_k} \tag{5.24}$$

$$0 \leq \frac{\tau_{\max}}{A_{\max}} - \frac{\tau_k}{A_k} \tag{5.25}$$

$$0 \leq \frac{A_k}{A_{\max}} - \frac{\tau_k}{\tau_{\max}}, \tag{5.26}$$

or $0 \leq h_{\mathrm{ts}}(\tau_k)$ for short. The constant $A_{\max}$ is a tuning variable. The constant $\tau_{\max}$ is fixed by $\tau_{\max} = \sigma_{\max} A_{\max}$.

Maximum Acceleration Constraint

An acceleration sensor is located on each kite, which is used as part of the localization procedure. Typical acceleration sensors only work up to certain limits, which are usually displayed in multitudes of the standard gravity g. To enforce this hardware constraint, we formulate

$$0 \leq g c_{\text{acc}} - \|\ddot{\mathbf{q}}_a\| \tag{5.27}$$

for all kites $a \in \mathcal{A}$ and with the constant $0 < c_{\text{acc}}$. For better scaling and to get rid of the nonlinearity of the norm, constraint (5.27) is reformulated as

$$\|\ddot{\mathbf{q}}_a\| \leq g c_{\text{acc}} \tag{5.28}$$

$$\ddot{\mathbf{q}}_a^{\mathsf{T}} \ddot{\mathbf{q}}_a \leq g^2 c_{\text{acc}}^2 \tag{5.29}$$

$$0 \leq 1 - \frac{\ddot{\mathbf{q}}_a^{\mathsf{T}} \ddot{\mathbf{q}}_a}{g^2 c_{\text{acc}}^2}, \tag{5.30}$$

or $0 \leq h_{\text{acc}}(\ddot{\mathbf{q}}_a)$ for short.

Anti-Collision Constraint

The kites must not collide during their flight, so we need to formulate an anti-collision constraint. The most straightforward way to do so is to enforce

$$0 \leq \|\mathbf{q}_{a_1} - \mathbf{q}_{a_2}\| - d_{\text{min}}, \tag{5.31}$$

where d_{min} is the minimum distance the kites should be apart at all times. This must hold for all combinations of $(a_1, a_2) \in \{(a_1, a_2) \mid a_1, a_2 \in \mathcal{A} \wedge a_1 \neq a_2\} = \mathcal{A}_{\text{com}}$. It makes sense to define d_{min} as a multiple of the kite's wing span b_a, resulting in $d_{\text{min}} = c_{\text{min}} b_a$ with the constant factor c_{min}. Formulating the constraint as proposed in equation (5.31) is however unwise, since it does not scale well and is not continuously differentiable at the origin due to the norm. Hence, the constraint is reformulated as

$$d_{\text{min}} \leq \|\mathbf{q}_{a_1} - \mathbf{q}_{a_2}\| \tag{5.32}$$

$$d_{\text{min}}^2 \leq (\mathbf{q}_{a_1} - \mathbf{q}_{a_2})^{\mathsf{T}} (\mathbf{q}_{a_1} - \mathbf{q}_{a_2}) \tag{5.33}$$

$$0 \leq \frac{(\mathbf{q}_{a_1} - \mathbf{q}_{a_2})^{\mathsf{T}} (\mathbf{q}_{a_1} - \mathbf{q}_{a_2})}{d_{\text{min}}^2} - 1, \tag{5.34}$$

or $0 \leq h_{\text{acol}}(\mathbf{q}_{a_1}, \mathbf{q}_{a_2})$ for short.

Terminal Position Constraint

To ensure the success of a landing trajectory, we constrain the terminal position of the nodes $\mathbf{q}_n$ to be within a certain distance to the ground-station. To do so, we formulate the constraint

$$\|\mathbf{q}_{p,\bar{m},n}\| - d_{\text{f},p} \leq 0 \tag{5.35}$$

for all $p \in \mathcal{N}$. The constant $d_{\mathrm{f},p}$ is the maximum terminal distance of node p from the ground-station. For the reasons that were already explained before, the constraint is reformulated as

$$0 \leq 1 - \frac{(\mathbf{q}_{p,\bar{m},n}^{\mathsf{T}} \mathbf{q}_{p,\bar{m},n})}{d_{\mathrm{f},p}^2}, \tag{5.36}$$

or $0 \leq h_{\mathrm{tp}}(\mathbf{q}_{p,\bar{m},n})$ for short.

Minimum Battery Energy Constraint

The minimum battery energy constraint enforces that the energy stored on the on-board battery at the any time must be above a certain threshold $E_{\mathrm{B,f}}$. This is expressed by the inequality

$$E_{\mathrm{B},i} - E_{\mathrm{B,f}} \geq 0, \tag{5.37}$$

for all $i \in \mathcal{I}$. The energy $E_{\mathrm{B},i}$ is modeled just as explained in section 4.5. When discretizing the equation (4.37), we arrive at

$$E_{\mathrm{B},\bar{i}} = \bar{E}_{\mathrm{B},0} + \frac{t_{\mathrm{f}}}{\bar{i}} \sum_{i=0}^{\bar{i}-1} \sum_{j=1}^{n} w_{\mathrm{q},j} \dot{E}_{\mathrm{B},i,j}, \quad \forall \bar{i} \in \mathcal{I}_- \tag{5.38}$$

$$E_{\mathrm{B},0} = \bar{E}_{\mathrm{B},0}, \tag{5.39}$$

where we use Radau quadrature to approximate the integral just as explained in section 2.2. Since we simulate a compromised on-board generator, we set $\dot{E}_{\mathrm{B,in},i,j} = 0$ for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$.

This approach only enforces the constraint on the last node of every collocation interval. In addition, it is not necessary to implement the battery energy and its derivative as states of the problem, which would generate a lot more constraints and optimization variables. If enforcing the constraint only once per collocation interval is not accurate enough, one can also add the battery energy and its derivative as additional states to the OCP with the simple ODE $\frac{\mathrm{d}}{\mathrm{d}t} E_{\mathrm{B}} = \dot{E}_{\mathrm{B}}$ and then enforce the constraint

$$E_{\mathrm{B},i,j} - E_{\mathrm{B,f}} \geq 0 \tag{5.40}$$

for all $i \in \mathcal{I}_-$ and $j \in \mathcal{J}$ as well as

$$E_{\mathrm{B},0,0} = N_{\mathrm{B}} \nu_{\mathrm{fr}} \nu_{\mathrm{c}} Q_{\mathrm{B}} U_{\mathrm{B}}. \tag{5.41}$$

Since we want to introduce the minimum battery energy constraint during the homotopy scheme, we reformulate equation (5.38) as

$$E_{\mathrm{B},\bar{i}}(\boldsymbol{\phi}) = E_{\mathrm{B},0} + \frac{t_{\mathrm{f}}}{\bar{i}} \sum_{i=0}^{\bar{i}-1} \sum_{j=1}^{n} w_{\mathrm{q},j} \dot{E}_{\mathrm{B},i,j} (1 - \nu), \quad \forall \bar{i} \in \mathcal{I}_- \tag{5.42}$$

with the homotopy parameter ν that corresponds to the compromised landing homotopy step (cf. section 5.3).

State and Control Bounds

Since the main tether length l_T is a distance, it must be bounded from below by

$$0 \leq l_\mathrm{T} \tag{5.43}$$

for the model to stay physical. Similarly, the height of all nodes must be positive since we assume that a vanishing height corresponds to ground level. For safety reasons, it seems reasonable to constrain the height to a value slightly above zero in order to account for uneven terrain. This is expressed as

$$z_\mathrm{min} \leq q_{z,k}, \tag{5.44}$$

with the minimum height z_min. The kites' lift coefficients $C_{\mathrm{L},a}$ and roll angle Ψ_a must be bounded by

$$C_\mathrm{L,min} \leq C_{\mathrm{L},a} \leq C_\mathrm{L,max} \tag{5.45}$$

and

$$\Psi_\mathrm{min} \leq \Psi_a \leq \Psi_\mathrm{min} \tag{5.46}$$

respectively for all $a \in \mathcal{A}$ in accordance with the kites' geometry and aerodynamics. Also, the control inputs $\mathbf{u}$ must be bounded by suitable

$$\dot{C}_\mathrm{L,min} \leq \dot{C}_{\mathrm{L},a} \leq \dot{C}_\mathrm{L,max} \tag{5.47}$$

$$\dot{\Psi}_\mathrm{min} \leq \dot{\Psi}_a \leq \dot{\Psi}_\mathrm{max} \tag{5.48}$$

$$\ddot{l}_\mathrm{T,min} \leq \ddot{l}_\mathrm{T} \leq \ddot{l}_\mathrm{T,max}, \tag{5.49}$$

which reflect the minimum and maximum possible actuation of the system for all $a \in \mathcal{A}$. To allow for a change in the variable bounds on $C_{\mathrm{L},a}$, Ψ_a, $\dot{C}_{\mathrm{L},a}$ and $\dot{\Psi}_a$ during the homotopy scheme, we use the homotopy parameter ν to define

$$C_\mathrm{L,max} = \nu C^+_\mathrm{L,max} + (1-\nu) C^-_\mathrm{L,max} \tag{5.50}$$

$$C_\mathrm{L,min} = \nu C^+_\mathrm{L,min} + (1-\nu) C^-_\mathrm{L,min}. \tag{5.51}$$

The superindex + indicates the variable bound before the switch, while the superindex − indicates the variable bound afterwards. The bounds on Ψ_a, $\dot{C}_{\mathrm{L},a}$ and $\dot{\Psi}_a$ are defined analogously. In the following, equations (5.43) to (5.49) are summarized as

$$\mathbf{x}_\mathrm{min}(\boldsymbol{\phi}) \leq \mathbf{x} \leq \mathbf{x}_\mathrm{max}(\boldsymbol{\phi}) \tag{5.52}$$

$$\mathbf{u}_\mathrm{min}(\boldsymbol{\phi}) \leq \mathbf{u} \leq \mathbf{u}_\mathrm{max}(\boldsymbol{\phi}) \tag{5.53}$$

for ease of notation. Furthermore, we introduce

$$[C^+_{\mathrm{L,min},a}, C^+_{\mathrm{L,max},a}, \Psi^+_{\mathrm{min},a}, \Psi^+_{\mathrm{max},a}]^\mathsf{T} = \mathbf{x}^+_{\delta,a} \tag{5.54}$$

$$[C^-_{\mathrm{L,min},a}, C^-_{\mathrm{L,max},a}, \Psi^-_{\mathrm{min},a}, \Psi^-_{\mathrm{max},a}]^\mathsf{T} = \mathbf{x}^-_{\delta,a} \tag{5.55}$$

$$[\dot{C}^+_{\mathrm{L,min},a}, \dot{C}^+_{\mathrm{L,max},a}, \dot{\Psi}^+_{\mathrm{min},a}, \dot{\Psi}^+_{\mathrm{max},a}]^\mathsf{T} = \mathbf{u}^+_{\delta,a} \tag{5.56}$$

$$[\dot{C}^-_{\mathrm{L,min},a}, \dot{C}^-_{\mathrm{L,max,a}}, \dot{\Psi}^-_{\mathrm{min},a}, \dot{\Psi}^-_{\mathrm{max},a}]^\mathsf{T} = \mathbf{u}^-_{\delta,a}. \tag{5.57}$$

The numerical values for the bounds introduced in this section can be seen in appendix A.2.

5.3. Homotopy Schedule

Rather than solving a single OCP to generate the desired emergency trajectories, we use a homotopy scheme and solve a number of OCPs one after the other until arriving at the solution. The homotopy building blocks introduced in this section are the initial homotopy step, the tracking homotopy step, the transition homotopy step, the nominal landing homotopy step, the compromised landing homotopy step and the final homotopy step. Within the current section, the force vectors $\mathbf{F}_{(\cdot)}$ do not have the indices i and j, because the forces refer to the DAE $\mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t_\mathrm{f}, \mathbf{p})$ that appears in the collocation constraints. A change in $\mathbf{f}$ obviously affects all collocation points $i \in \mathcal{I}$, $j \in \mathcal{J}_-$.

Initial Homotopy Step

The initial and the tracking homotopy step are used to generate a feasible initial guess out of a crude one. As explained in section 2.4, we employ homotopy parameters inside the model of the system dynamics. Specifically, we use the homotopy parameter γ to extend the aerodynamic forces $\mathbf{F}_{\mathrm{A},a} = \mathbf{F}_{\mathrm{L},a} + \mathbf{F}_{\mathrm{D},a}$ in the model with

$$\begin{aligned} \mathbf{F}_{\mathrm{A},a,+}(\mathbf{p}) &= (1-\gamma)(\mathbf{F}_{\mathrm{L},a} + \mathbf{F}_{\mathrm{D},a}) + \gamma(\mathbf{F}_{\mathrm{L},a} + \mathbf{F}_{\mathrm{D},a} + \mathbf{F}_{\mathrm{F},a}) \\ &= \mathbf{F}_{\mathrm{L},a} + \mathbf{F}_{\mathrm{D},a} + \gamma\mathbf{F}_{\mathrm{F},a}, \end{aligned} \tag{5.58}$$

for all kites $a \in \mathcal{A}$, where $\mathbf{F}_{\mathrm{F},a}$ are the fictitious forces acting on kite a. In accordance with the notation of section 2.4, we use $\boldsymbol{\theta} = \mathbf{F}_\mathrm{F}$, where $\mathbf{F}_\mathrm{F} \in \mathbb{R}^{3N_\mathrm{K}}$ is a concatenation of the vectors $\mathbf{F}_{\mathrm{F},a}$ for all $a \in \mathcal{A}$. This means that for $\gamma = 1$, which is the initial value for all bounded homotopy parameters, the nonlinearity of the aerodynamic forces are still present in the model, but the solver can choose an appropriate $\mathbf{F}_{\mathrm{F},a}$ for the trajectory to be feasible during the optimization. Empirically, choosing the starting function of the homotopy to be $\mathbf{F}_{\mathrm{A},a} + \mathbf{F}_{\mathrm{F},a}$ has worked better than choosing it as $\mathbf{F}_{\mathrm{F},a}$. Intuitively, it makes sense that the homotopy can be solved faster if the starting function is more similar to the target function. However, further research needs to be conducted with respect to how the starting functions need to be chosen for a given homotopy.

Additionally, we set $\tau_0 = \tau_\mathrm{a}$, $\tau_\mathrm{f} = \tau_\mathrm{b}$ and $r_\mathrm{tol} = r_\mathrm{a}$ during the initial homotopy step.

Tracking Homotopy Step

In the tracking homotopy step, the bounded homotopy parameter γ and the unbounded homotopy parameter $\mathbf{F}_\mathrm{F}$ are used to change from fictitious forces $\mathbf{F}_\mathrm{F}$ to aerodynamic forces $\mathbf{F}_\mathrm{A}$ in the dynamics. By doing so, a crude initial guess is used as a reference to generate a feasible one that resembles the reference as much as possible due to the

tracking cost T. This can be especially interesting since the crude initial guess need not necessarily include good guesses for the controls $\mathbf{u}$, the algebraic variables $\boldsymbol{\lambda}$ or even parts of the states $\mathbf{x}$. Assuming that most states are chosen consistently, the rest get filled in by the homotopy process. This is very useful since, from experience, it suffices to find a consistent initial guess for $\mathbf{q}$, $\dot{\mathbf{q}}$, l_T and $\dot{l}_\mathrm{T}$ that are much easier to formalize in algebraic expressions than the rest of $\mathbf{x}$, $\mathbf{u}$ and $\boldsymbol{\lambda}$. With the notation of section 2.4, this homotopy step can be formalized as

$$\text{tracking:} \quad \mathbf{F}_\mathrm{FA} \xrightarrow[\mathbf{F}_\mathrm{F}]{\gamma} \mathbf{F}_\mathrm{A}, \tag{5.59}$$

where $\mathbf{F}_\mathrm{FA} = \mathbf{F}_\mathrm{F} + \mathbf{F}_\mathrm{A}$. Beginning with the tracking homotopy step, we set $\tau_0 = \tau_\mathrm{b}$ and $\tau_\mathrm{f} = \tau_\mathrm{b}$.

Transition Homotopy Step

The transition homotopy step is used to switch from the initial tracking problem to the transition problem. Since the necessary initial and terminal constraints are already enforced during the initial and the tracking homotopy step, we only switch from the tracking cost T to the transition cost K. This is done by using the homotopy parameter υ. The transition homotopy step is expressed as

$$\text{transition:} \quad T \xrightarrow{\upsilon} K. \tag{5.60}$$

Nominal Landing Homotopy Step

In the nominal landing homotopy step, we use the homotopy parameter η to switch from the tracking problem of the previous homotopy step to the landing problem that we actually want to solve. We do this by switching the cost from T to L, thereby no longer tracking the reference but penalizing the terminal position and velocity. Since we use the solution of the previous homotopy step as an initial guess, we can assume that the solver converges to a local minimum that is reasonably close to the tracking reference. However, since $\eta T = 0$ at the end of this homotopy step, there is no gradient in the cost that forces the solution towards the reference trajectory. The homotopy step is formalized as

$$\text{nominal landing:} \quad T \xrightarrow{\eta} L. \tag{5.61}$$

Compromised Landing Homotopy Step

In order to model a compromised landing, we use the homotopy parameter ν to change the bounds on the states $C_{\mathrm{L},a}$, Ψ_a, on the controls $\dot{C}_{\mathrm{L},a}$ and $\dot{\Psi}_a$ or on the on-board battery energy $E_{\mathrm{B},a}$ of one kite $a \in \mathcal{A}$. The bounds are changed according to the emergency scenario that is being modeled in the OCP. It also needs to be kept in mind that when making the bounds on $C_{\mathrm{L},a}$ and Ψ_a more restrictive, it is not necessarily possible anymore to enforce

$$\mathbf{0} = [C_{\mathrm{L},a,0,0}, \Psi_{a,0,0}]^\mathsf{T} - \left[\hat{C}_{\mathrm{L},a,0}(\xi_0), \hat{\Psi}_{a,0}(\xi_0)\right]^\mathsf{T} \tag{5.62}$$

as part of the initial constraint

$$\mathbf{0} = \hat{\mathbf{x}}_0(\xi_0) - \mathbf{x}_{0,0}. \tag{5.63}$$

In these cases we must omit $C_{\mathrm{L},a}$ and Ψ_a from the initial constraint by reformulating the constraint as

$$\mathbf{0} = \hat{\mathbf{x}}_0^-(\xi_0) - \mathbf{x}_{0,0}^-, \tag{5.64}$$

where the superindex $-$ indicates that $C_{\mathrm{L},a}$ and Ψ_a are not a part of the vectors. We formalize the compromised landing homotopy step as

$$\text{compromised landing:} \quad \underbrace{[b_{1,\min}, b_{1,\max}]}_{\mathbf{b}_1} \xrightarrow{\nu} \underbrace{[b_{2,\min}, b_{2,\max}]}_{\mathbf{b}_2}, \tag{5.65}$$

where $\mathbf{b}_1$ and $\mathbf{b}_2$ are the bounds whose exact shape depends on the kind of compromised landing that is being modeled.

Tether Drag Homotopy Step

We use the homotopy parameter τ to extend the tether drag forces $\mathbf{F}_{\mathrm{TD},k}$ to

$$\mathbf{F}_{\mathrm{TD}+,k}(\boldsymbol{\phi}) = \tau \mathbf{F}_{\mathrm{TD},k}^{\mathrm{t}} + (1-\tau)\mathbf{F}_{\mathrm{TD},k}^{\mathrm{2nd}}, \quad \forall k \in \mathcal{T}, \tag{5.66}$$

where the superindices t and 2nd denote the trivial and a second tether drag model derived in section 4.4. This makes it possible to first solve a number of homotopy steps for the trivial tether model before switching to a more nonlinear model, making all previous homotopy steps easier to solve. Here, we refer to the trivial, projected and equivalence tether drag forces as $\mathbf{F}_{\mathrm{TD}}^{\mathrm{t}}$, $\mathbf{F}_{\mathrm{TD}}^{\mathrm{p}}$ and $\mathbf{F}_{\mathrm{TD}}^{\mathrm{e}}$ respectively. During this homotopy step, the cost function stays unchanged but for the cost related to the homotopy parameter τ. This homotopy step cannot only be used to switch from the trivial model to the projected model but also to other models of arbitrary complexity. The more complex the tether model is, the more useful it is to introduce the model in a later homotopy step such that the previous steps do not have to deal with the model's nonlinearities. The tether drag homotopy step is expressed as

$$\text{tether:} \quad \mathbf{F}_{\mathrm{TD}}^{\mathrm{t}} \xrightarrow{\tau} \mathbf{F}_{\mathrm{TD}}^{\mathrm{2nd}}. \tag{5.67}$$

Final Homotopy Step

In the final homotopy step, all constraints and the cost function of the OCP stay the same. However, the problem is now solved up to a small barrier parameter τ_{c} and error tolerance r_{b} as explained in section 2.4. The final homotopy step introduces additional nonlinearity into the problem by approximating the inequality constraints more accurately. It is used to solve the OCP to the desired degree of accuracy.

5.4. Initial Guess Generation

Finding a feasible initial guess is non-trivial for a dual-kite AWES. Hence, we use the homotopy strategy that is outlined in sections 2.4 and 5.3. However, this homotopy strategy must still be initialized with a reasonably good initial guess, even though it need not be feasible. When looking for an initial guess $\mathbf{v}^0 = [\mathbf{x}^{0\mathrm{T}}, \mathbf{u}^{0\mathrm{T}}, \boldsymbol{\lambda}^{0\mathrm{T}}, \mathbf{p}^{0\mathrm{T}}]^\mathsf{T}$, we have the following criteria in mind:

- The initial guess must be consistent, meaning that $\left[\mathbf{g}(\mathbf{x}^0_{i,j}), \dot{\mathbf{g}}(\mathbf{x}^0_{i,j})\right]^\mathsf{T} = \mathbf{0}$ must hold for all $i \in \mathcal{I}$, $j \in \mathcal{J}$ and not only for $[i, j] = [0, 0]$ as enforced in equation (4.6).
- All inequality constraints from section 5.2.2 must hold for $\mathbf{v}^0$.
- All equality constraints from section 5.2.1 most hold for $\mathbf{v}^0$ with the exception of the collocation constraints. The latter depend nonlinearly on $\mathbf{u}$ and $\boldsymbol{\lambda}$ for which it is very hard to find a good initial guess. Also, the parameterized terminal and initial constraints must only hold for $\mathbf{q}$, l_T and $\dot{l}_\mathrm{T}$.

To make sure that these criteria are met, we generate initial guesses as functions of the OCP's parameterized initial and terminal conditions. The idea is to select an initial and terminal point for the trajectory and then interpolate the two points in a way that ensures the criteria are all met. The initial guesses for transition and landing trajectories differ in that the transition trajectory is constrained by both the initial and terminal condition, while the landing is only constrained by the initial condition.

5.4.1. Initial and Terminal Points

For both transition and nominal landing trajectories, it makes the most sense to start at the point where the tether velocity $\dot{\hat{l}}_{\mathrm{T},0}$ of the parameterized initial condition $\hat{\mathbf{x}}_0$ is minimal. This happens during the pumping cycle's reel-in phase. We denote this initial point as

$$\mathbf{x}^0_{0,0} \in \left\{ \hat{\mathbf{x}}_0(\bar{\xi}_0) \mid \dot{\hat{l}}_{\mathrm{T},0}(\bar{\xi}_0) \leq \dot{\hat{l}}_{\mathrm{T},0}(\xi_0),\ \bar{\xi}_0 \in [0,1],\ \forall \xi_0 \in [0,1] \right\}. \tag{5.68}$$

The terminal point for the transition trajectories is chosen analogously as the point on $\hat{\mathbf{x}}_\mathrm{f}$ where $\dot{\hat{l}}_{\mathrm{T,f}}$ is minimal. Hence, we define $\mathbf{x}^0_{\bar{m},n}$ analogously to $\mathbf{x}^0_{0,0}$. For compromised landing trajectories, the initial point on $\hat{\mathbf{x}}_0$ is already fixed by ξ_0, so we set

$$\mathbf{x}^0_{0,0} = \hat{\mathbf{x}}_0(\xi_0), \tag{5.69}$$

for a given ξ_0 that is not a decision variable of the OCP. The terminal point $\mathbf{x}^0_{\bar{m},n}$ for all landing trajectories is not constrained by a $\hat{\mathbf{x}}_\mathrm{f}$ as it is the case for the transition trajectories. We choose $\mathbf{x}^0_{\bar{m},n}$ for the landing trajectories by preserving all tether vectors $\hat{\mathbf{e}}_k$ of $\mathbf{x}^0_{0,0}$, but reducing the AWES's main tether length l^0_T such that the inequality constraint (5.44) holds.

5.4.2. Consistent Interpolation

Now that we have generated tuples of $(\mathbf{x}^0_{0,0}, \mathbf{x}^0_{\bar{m},n})$ according to the respective trajectory, the interpolation procedure does not depend on the type of the trajectory anymore but only on the tuple. First we construct a coordinate system consisting of the three unit vectors $\mathbf{e}_{||}$, $\mathbf{e}_+$ and $\mathbf{e}_\perp$, which are defined as follows: The vector $\mathbf{e}_{||}$ is the normalized bisection of $\hat{\mathbf{e}}_2$ and $\hat{\mathbf{e}}_3$, which is expressed as

$$\mathbf{e}_{||} = \left(\hat{\mathbf{e}}_2 + \frac{1}{2}(\hat{\mathbf{e}}_3 - \hat{\mathbf{e}}_2)\right)\left(\|\hat{\mathbf{e}}_2 + \frac{1}{2}(\hat{\mathbf{e}}_3 - \hat{\mathbf{e}}_2)\|\right)^{-1}. \tag{5.70}$$

The vector $\mathbf{e}_+$ is orthogonal to $\mathbf{e}_{||}$ and lies in the surface that contains $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$. It is defined as

$$\mathbf{e}_+ = \frac{\hat{\mathbf{e}}_3 - \hat{\mathbf{e}}_2}{\|\hat{\mathbf{e}}_3 - \hat{\mathbf{e}}_2\|}. \tag{5.71}$$

The third vector $\mathbf{e}_\perp$ is defined as the normalized cross product of $\mathbf{e}_{||}$ and $\mathbf{e}_+$. Hence, $\mathbf{e}_{||}$, $\mathbf{e}_+$ and $\mathbf{e}_\perp$ define an orthonormal coordinate system. The coordinate system's origin is located at the center node $\mathbf{q}_1$.

The movement of the center node is parameterized in time as

$$\mathbf{q}_1(t) = \mathbf{q}^0_{1,0,0} + s_0(t)(\mathbf{q}^0_{1,\bar{m},n} - \mathbf{q}^0_{1,0,0}), \tag{5.72}$$

where $s_0(t) \in [0,1]$ is a time dependent interpolation parameter. We define $s_0(t)$ in such a way that it corresponds to a given function $l^0_{\mathrm{T}}(t)$ by solving the system of equations

$$l^0_{\mathrm{T}}(t)\hat{\mathbf{e}}_1(t) = \mathbf{q}^0_{1,0,0} + s_0(t)(\mathbf{q}^0_{1,\bar{m},n} - \mathbf{q}^0_{1,0,0}) \tag{5.73}$$

$$\hat{\mathbf{e}}_1(t)^{\mathsf{T}}\hat{\mathbf{e}}_1(t) = 1, \tag{5.74}$$

which is equivalent to solving the quadratic equation

$$\begin{aligned} 0 = {} & s_0^2(t)(\mathbf{q}^0_{1,\bar{m},n} - \mathbf{q}^0_{1,0,0})^{\mathsf{T}}(\mathbf{q}^0_{1,\bar{m},n} - \mathbf{q}^0_{1,0,0}) \\ & + 2s_0(t){\mathbf{q}^0_{1,0,0}}^{\mathsf{T}}(\mathbf{q}^0_{1,\bar{m},n} - \mathbf{q}^0_{1,0,0}) \\ & + ({\mathbf{q}^0_{1,0,0}}^{\mathsf{T}}\mathbf{q}^0_{1,0,0}) - {l^0_{\mathrm{T}}}^2(t). \end{aligned} \tag{5.75}$$

For the given values of $l_{\mathrm{T}}(t)$ that are of interest for the initial guess generation, exactly one of the solutions of the quadratic equation (5.75) lies in the interval $[0,1]$ and is therefore the one we are looking for.

The function $l_{\mathrm{T}}(t)$, on which the parameterization of $\mathbf{q}_1(l_{\mathrm{T}}(t))$ depends, must be chosen such that the following boundary conditions are met:

$$l^0_{\mathrm{T}}(0) = l^0_{\mathrm{T},0,0} \tag{5.76}$$

$$l^0_{\mathrm{T}}(t^0_{\mathrm{f}}) = l^0_{\mathrm{T},\bar{m},n} \tag{5.77}$$

$$\dot{l}^0_{\mathrm{T}}(0) = \dot{l}^0_{\mathrm{T},0,0} \tag{5.78}$$

$$\dot{l}^0_{\mathrm{T}}(t^0_{\mathrm{f}}) = \dot{l}^0_{\mathrm{T},\bar{m},n} \tag{5.79}$$

To be able to enforce all the boundary conditions, $l_{\mathrm{T}}^0(t)$ must be at least cubic. Hence, we define

$$l_{\mathrm{T}}^0(t) = \frac{1}{6}c_3 t^3 + \frac{1}{2}c_2 t^2 + c_1 t + c_0, \tag{5.80}$$

where the constants c_0 to c_3 are uniquely defined by the boundary conditions (5.76) – (5.79). The initial guess for the tether velocity $\dot{l}_{\mathrm{T}}^0$ is derived by differentiating l_{T}^0 with respect to time. Note that a cubic tether length implies that the tether jerk is constant.

The coordinate system consisting of $\mathbf{e}_{||}$, $\mathbf{e}_+$ and $\mathbf{e}_\perp$ is used to parameterize the positions of $\mathbf{q}_2(t)$ and $\mathbf{q}_3(t)$ using spherical coordinates where ϕ denotes the azimuth and θ the elevation. The kite nodes' positions $\mathbf{q}_a(t)$ for $a \in \mathcal{A}$ are then parameterized as

$$\begin{aligned}\mathbf{q}_a(t) = \mathbf{q}_1(t) &\\ + (\cos(\phi_a(t))\cos(\theta_a(t))\mathbf{e}_{||} &\\ + \sin(\phi_a(t))\cos(\theta_a(t))\mathbf{e}_+ &\\ + \sin(\theta_a(t))\mathbf{e}_\perp) l_{\mathrm{s}}.&\end{aligned} \tag{5.81}$$

The expression for $\phi_a(t)$ is derived by linear interpolation between $\phi_{a,0,0}^0$ and $\phi_{a,\bar{m},n}^0$ as

$$\phi_a(t) = \phi_{a,0,0}^0 + \frac{t}{t_{\mathrm{f}}^0}(\phi_{a,\bar{m},n}^0 - \phi_{a,0,0}^0). \tag{5.82}$$

The elevation $\theta_a(t)$ is defined analogously. The constants $\phi_{a,0,0}^0$, $\phi_{a,\bar{m},n}^0$, $\theta_{a,0,0}^0$ and $\theta_{a,\bar{m},n}^0$ are derived by

$$\theta_{a,0,0}^0 = \arcsin\left(\mathbf{e}_{||}^{\mathsf{T}}\hat{\mathbf{e}}_a(0)\right) \tag{5.83}$$

$$\theta_{a,\bar{m},n}^0 = \arcsin\left(\mathbf{e}_{||}^{\mathsf{T}}\hat{\mathbf{e}}_a(t_{\mathrm{f}}^0)\right) \tag{5.84}$$

$$\phi_{a,0,0}^0 = \operatorname{arctan2}\left((l_{\mathrm{s}}\hat{\mathbf{e}}_a(0))^{\mathsf{T}}\mathbf{e}_+, (l_{\mathrm{s}}\hat{\mathbf{e}}_a(0))^{\mathsf{T}}\mathbf{e}_{||}\right) \tag{5.85}$$

$$\phi_{a,\bar{m},n}^0 = \operatorname{arctan2}\left((l_{\mathrm{s}}\hat{\mathbf{e}}_a(t_{\mathrm{f}}^0))^{\mathsf{T}}\mathbf{e}_+, (l_{\mathrm{s}}\hat{\mathbf{e}}_a(t_{\mathrm{f}}^0))^{\mathsf{T}}\mathbf{e}_{||}\right). \tag{5.86}$$

The choice of $\mathbf{e}_{||}$, $\mathbf{e}_+$ and $\mathbf{e}_\perp$ makes sure that $\theta_{a,0,0}^0 = 0$ for all $a \in \mathcal{A}$, which ensures that that $|\theta_{a,0,0}^0 - \theta_{a,\bar{m},n}^0| \leq 2\pi$. The initial guess for the node velocities $\dot{\mathbf{q}}_n^0$ can be derived by differentiating $\mathbf{q}_n^0$ with respect to time. The initial guess for Ψ_a^0 and $C_{\mathrm{L},a}^0$ is simply set to zero.

After defining $\mathbf{x}^0(t)$ as a continuous function of t, the values for $\mathbf{x}_{i,j}^0$ are generated by

$$\mathbf{x}_{i,j}^0 = \mathbf{x}^0(t_{i,j}), \tag{5.87}$$

for all $i \in \mathcal{I}$, $j \in \mathcal{J}$, where $t_{i,j}$ is defined as in section 2.2. Lastly, the values for $\mathbf{u}^0$ are all set to zero and $\boldsymbol{\lambda}^0$ to one, since we do not have a better initial guess for them. We set $\boldsymbol{\lambda}^0 \geq \mathbf{0}$ to ensure that constraint (5.21) holds. The initial guess for the homotopy parameters is derived by setting $\boldsymbol{\theta}^0 = \mathbf{0}$ and initializing $\boldsymbol{\phi}^0$ according to the corresponding bounds of homotopy step $\mathcal{P}_l$ as explained in section 2.4. The initial guess for the terminal time t_{f} is computed by

$$t_{\mathrm{f}}^0 = \frac{l_{\mathrm{T},0,0}^0 - l_{\mathrm{T},\bar{m},n}^0}{v_{\mathrm{T}}}, \tag{5.88}$$

where v_{T} is chosen as a sensible reel-in speed for the main tether.

5.5. Numerical Issues

During the implementation of the optimization problems, a number of numerical issues have come up. Specifically, these are the infeasibility of small parts of the parameterized trajectories, the need for an acceleration regularization as well as the tuning of meta-parameters of the homotopy procedure and the addition of a "stagger" distance to the tracking trajectory.

5.5.1. Infeasibility of Parameterized Trajectories

In general, the trajectory parameterization using `CasADi`-splines is very accurate in each of the states in $\mathbf{x}$. Nevertheless, the trajectory $\hat{\mathbf{x}}_0(\xi_0)$ is infeasible for some $\xi_0 \in [0, 1]$. The reason for this is illustrated in Fig. 5.1. The plot on the left displays a comparison of the approximated C_L for an example pumping trajectory and the real value for C_L. The parameterization looks almost identical to the real trajectory and has indeed a lookup-table error of around $1 \cdot 10^{-12}$.

The plot on the right is a zoomed-in detail of the left plot. It shows a small overshoot phenomenon when moving to or from the maximum bound of C_L (dashed line). Even though the effect is small with an order of magnitude of $1 \cdot 10^{-3}$, it is large enough for the solver to consider those points infeasible for low barrier parameters. The same effect can be observed for all states that have active bounds during the trajectory, but is in practice only relevant for C_L and Ψ. For $\xi \in [0, 1]$ this means that some points in the interval, which could otherwise be optimal, are infeasible for purely numerical reasons. When "accidentally" constraining ξ to an infeasible point, the solver is unable to solve the problem. Hence, we soften the relevant bounds on C_L and Ψ by a small $\epsilon = 1 \cdot 10^{-3}$. Since ϵ is so small, it has otherwise no effect on the solution of the OCP.

5.5.2. Acceleration Regularization

Heuristically, a regularization on the node accelerations has proven to increase convergence and give solutions with better invariants. In [23], a similar regularization directly on the kite forces of a single kite AWES is employed to generate smoother landing trajectories. In order to quantify this phenomenon, we observe the numerical performance of a nominal landing trajectory (for more details, cf. chapter 7) while varying the regularization on the node accelerations. Fig. 5.2 shows a comparison of the numerical performance for different values of $\hat{R}_{\ddot{\mathbf{q}}}$. The values of $\hat{R}_{\ddot{\mathbf{q}}}$ are plotted logarithmically with $\hat{R}_{\ddot{\mathbf{q}}} = 0$ corresponding to the value of $-\infty$.

The number of iterations decreases drastically when increasing $\hat{R}_{\ddot{\mathbf{q}}}$. Between $\hat{R}_{\ddot{\mathbf{q}}} = 0$ and $\hat{R}_{\ddot{\mathbf{q}}} = 1 \cdot 10^{-1}$, the number of iterations decreases by 88 % from 1135 to 136 iterations. The optimization time similarly decreases by 82 %. The time needed to construct the OCP does not change with $\hat{R}_{\ddot{\mathbf{q}}}$. The maximum invariant decreases by 90 % when introducing a regularization of $\hat{R}_{\ddot{\mathbf{q}}} = 1 \cdot 10^{-6}$ instead of $\hat{R}_{\ddot{\mathbf{q}}} = 0$. When we increase $\hat{R}_{\ddot{\mathbf{q}}}$

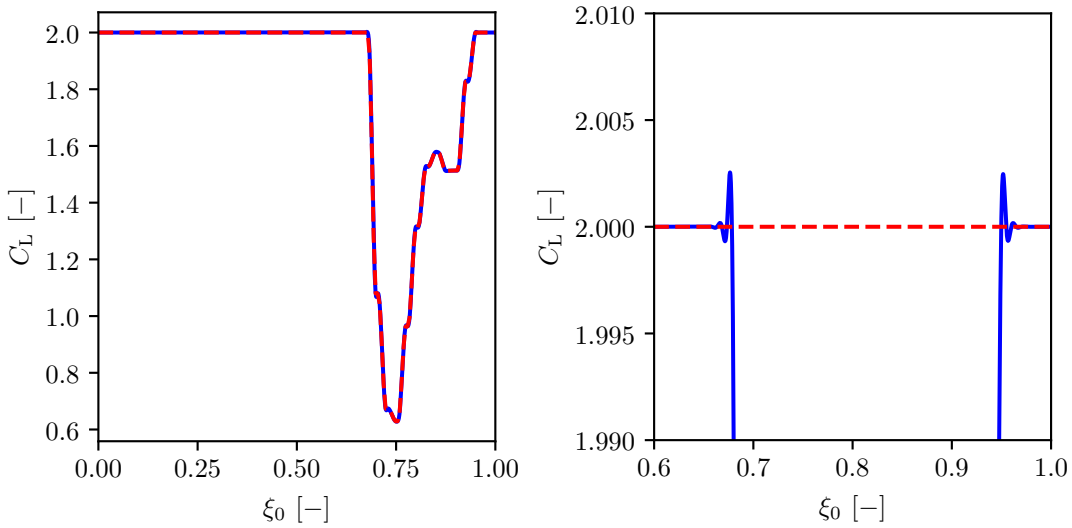

Figure 5.1.: Infeasibility of parameterized trajectories. Left: Comparison of spline approximation (blue) and actual curve (red) for C_L. Right: Zoomed-in detail of overshoot phenomenon with dashed line indicating the bound on C_L.

further, the maximum invariants vary between 4 % and 17 % of the original value. In the example trajectory shown in Fig. 5.2, a choice of $\hat{R}_{\ddot{\mathbf{q}}} = 1 \cdot 10^{-1}$ or $1 \cdot 10^{-2}$ is reasonable, depending on whether one prioritizes optimization speed or the size of the maximum invariants.

Large invariants are a sign of discretization errors in the DAE of the system. Large node accelerations mean that a large number of collocation intervals are necessary to make the discretization sufficiently accurate. Regularizing the node accelerations has the effect that we can discretize the DAE with fewer collocation intervals without increasing the invariants. Of course the discretization error, and with it the invariants, can also be decreased by increasing the number of collocation intervals. This however comes at the price of increasing the size of the problem. Therefore, we choose to fix the number of collocation intervals at a reasonable number and add a regularization on the node accelerations.

5.5.3. Homotopy Meta-Parameter Tuning

The success of the homotopy strategy (cf. section 2.4) depends on the tuning of the meta-parameters involved. Especially important is the choice of τ_b, which determines how well the inequalities are approximated during the intermediate homotopy steps. If τ_b is too small, the solver spends time on nonlinearities of the intermediate steps that do not appear in the final homotopy step. If τ_b is chosen too large, then the approximation of the inequality constraints is so crude in the intermediate homotopy steps that their

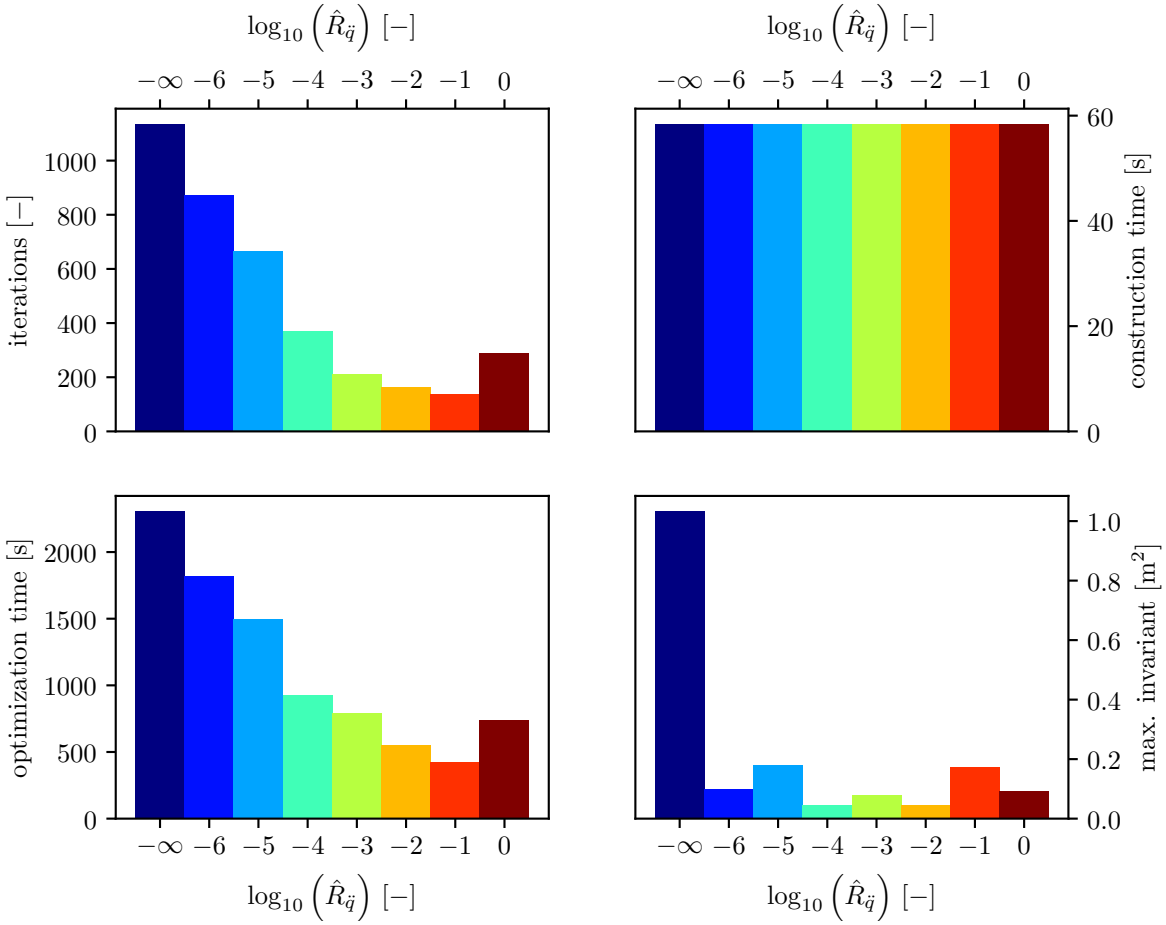

Figure 5.2.: Acceleration Regularization. Comparison of the number of iterations, NLP construction time, optimization time and maximum invariants for different values of $\hat{R}_{\ddot{\mathbf{q}}}$.

solutions are not good initial guesses for the next homotopy step anymore.

This phenomenon is visualized in Fig. 5.3. As an example, it shows the solution of the nominal landing homotopy step of a nominal landing trajectory (cf. chapter 7) for two different values of τ_b. The pumping cycle used as parameterized initial condition is displayed in black. For $\tau_\mathrm{b} = 1 \cdot 10^{-5}$ (blue), the solver converges to a sensible initial guess for a landing trajectory. For $\tau_\mathrm{b} = 1 \cdot 10^{-2}$ (red), the optimal landing trajectory first increases the kites' height to over 500 m before landing. This trajectory is obviously not a sensible solution and therefore also not a good initial guess for the next homotopy step. Hence, τ_b must be tuned in such a way that all intermediate solutions of the homotopy procedure are sensible.

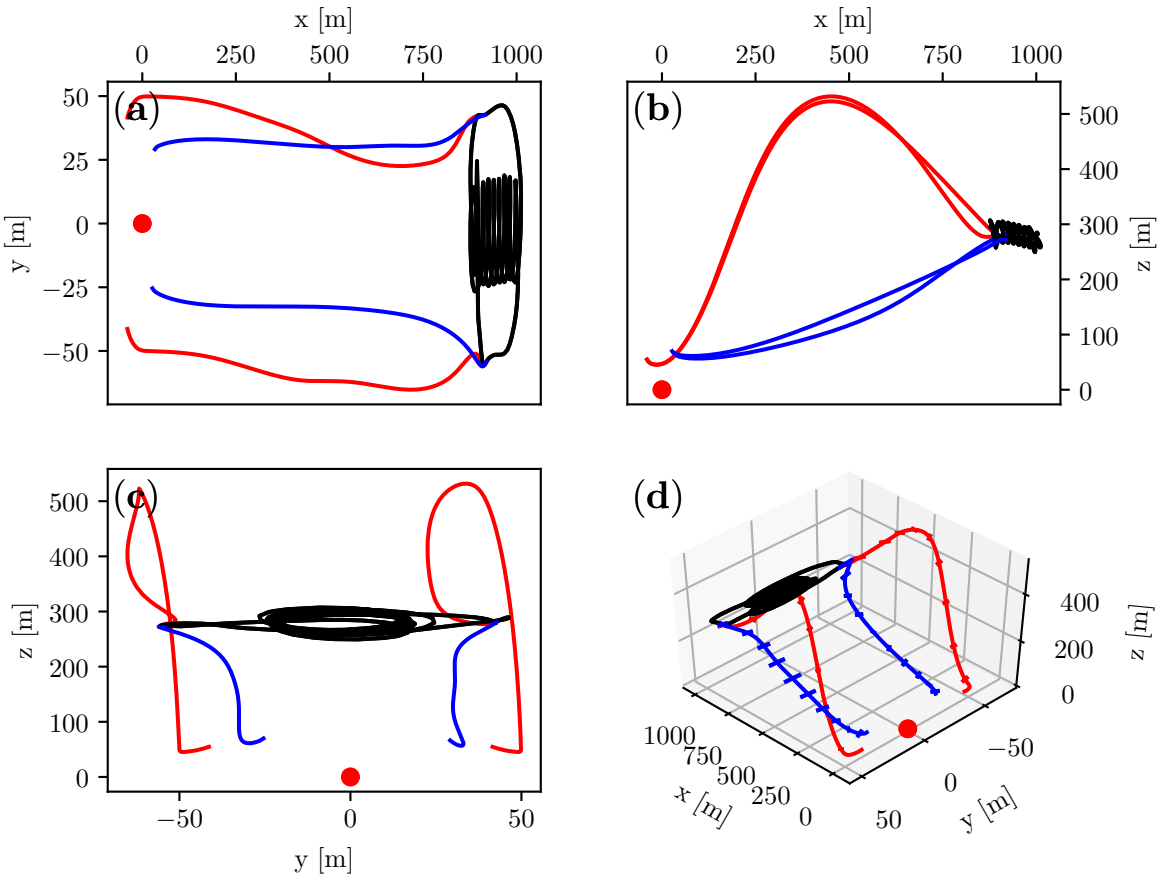

Figure 5.3.: Homotopy meta-parameter tuning. Comparison of solutions after nominal landing homotopy step for $\tau_\mathrm{b} = 1 \cdot 10^{-2}$ (red) and $1 \cdot 10^{-5}$ (blue). Initial trajectory displayed in black. Position of the ground-station represented by a red dot.

5.5.4. Stagger Distance

It has proven useful within the tracking homotopy step to add a "stagger" distance between the initial guess $\mathbf{x}^0$ and the tracking reference $\mathbf{x}_\mathrm{ref}$. This means that we do not set $\mathbf{x}_\mathrm{ref} = \mathbf{x}^0$, but add a small "stagger" distance d_s to all of the kites' positions $\mathbf{q}_k$ in the direction of their corresponding tethers. Specifically, we set

$$\mathbf{q}_{p,\mathrm{ref}} = \mathbf{q}_p^0 + d_\mathrm{s}\hat{\mathbf{e}}_p, \quad \forall p \in \mathcal{N}, \tag{5.89}$$

where $\hat{\mathbf{e}}_k$ is defined just like in equation (4.15). This approach is motivated by the fact that the small distance d_s between $\mathbf{x}^0$ and $\mathbf{x}_\mathrm{ref}$ encourages the solver to find trajectories with positive tether stress in the tracking homotopy step. With this, we can make sure that the initial guess produces sensible tether stress values. Adding a "stagger" distance

has empirically proven to improve convergence in the tracking homotopy step. We choose a value of $d_\mathrm{s} = 0.1$ m.

In this chapter, we have presented the modular building blocks that can now be used to construct OCP formulations. We have listed the components of the cost function as well as the equality and inequality constraints. Also, we have used the notation of section 2.4 to introduce a number of homotopy steps that will be used to solve the OCPs in this thesis. Lastly, we gave a summary of different numerical issues that have come up during the course of this thesis. With the help of the preliminary work done in this chapter, we can now go on to formulate OCPs and homotopy schemes in a precise and readable fashion in chapters 7 and 8. Before that however, we will elaborate on how the framework is implemented in the `python` toolbox `AWEbox`.

6. Software Implementation

The procedures for trajectory optimization in this thesis are implemented as part of new features for the `python` package `AWEbox`, which provides optimization procedures for various kinds of multi-kite AWES related problems. This chapter gives some insight into how the problems discussed in this thesis have been implemented into `AWEbox` and which design choices were made in the course thereof. To do so, section 6.1 gives an overview of `AWEbox`'s functionalities and summarizes all the contributions made to `AWEbox` during the scope of this thesis. Then section 6.2 gives some more insights about how the toolbox `CasADi` is used within the `AWEbox` to formulate NLPs and what solvers where chosen as the numerical backend for `AWEbox`.

6.1. `AWEbox`

The `AWEbox` package can be used to formulate multi-kite AWES related optimization problems within a very high level framework. The user simply chooses which optimization problems should be solved and leaves the rest to `AWEbox`. With this, AWE engineers will be able to assess their design concepts and find suitable trajectories without having to deal with the numerical tuning that is usually necessary to solve such optimization problems. Possible optimization problems that can be solved in `AWEbox` include the computation of power cycles, starting and landing trajectories, emergency trajectories or transition trajectories. It is also possible to optimize system parameters of a multi-kite AWES, like the length or diameter of the secondary tethers. The emergency response strategies themselves as discussed in chapter 3 are not part of `AWEbox`.

The basic structure of the `AWEbox` is shown in Fig. 6.1. The schematic is separated into two parts, each depicting one of the two levels of abstraction used in the `AWEbox`. The first level of abstraction is the `trial`. A single `trial` corresponds to one OCP formulation that can be solved to yield an optimal trajectory. The next level of abstraction is the `sweep`. A single `sweep` can contain an arbitrary number of `trial` objects. One uses `sweep` objects to easily construct and solve a number of `trial`s. Once a `sweep` is created by the user, it takes the user specified `sweep options` and uses it to build the desired `trial` objects. In the `sweep options`, the user can specify a list of parameters that should be varied during the `sweep`. Once all of the `trial`s are built, `sweep` executes its `run` method with the `trial` objects as an input. This returns the `solutions` that contain the optimal solutions and a number of numerical and system parameters of each `trial` that were produced during the optimization process. The `solutions` can then be passed

to a `visualization` object to generate plots.

The lower part of Fig. 6.1 shows what happens in each `trial` while the `build` and `run` methods are called by `sweep`. Each `trial` that is built uses its associated `options`, `model`, `formulation`, `nlp` and `optimization` objects. The `options` object contains all preferences regarding the `trail`'s configuration, e.g. in terms of system architecture, choice of model, or parameter values. The `model` object handles the construction of the dynamical system equations based on the settings set in `options`. The `formulation` object is used to make choices about the general layout of the problem formulation, e.g. whether a landing trajectory should be computed or a power cycle. The `nlp` object then constructs an NLP out of the `model` and the `formulation`, applying the chosen discretization procedure and setting up the objective function and constraints of the given NLP. Lastly, the `optimization` object sets up the `trial`'s homotopy schedule. Once this step is completed, the `trial` is considered built. During the `build` method of `sweep`, this procedure is performed for each individual `trial`. Within the `run` method of `sweep`, each of the built `trial` objects is solved using its `optimize` method to yield a `solution` object.

The `AWEbox` package is designed to separate the user from the numerics while still providing enough freedom to customize the user's problem formulation. Since `AWEbox` will be released with an open-source license (at an unspecified later date), users can also implement their own problem formulations in case they are not provided by `AWEbox` or edit existing ones. The code is structured in a generic and compartmentalized way that facilitates the exchanging of parts of the problem formulation, e.g. the implementation of new kite models or new discretization methods. An example of the work-flow for using `AWEbox` is presented in the appendix B.1. The code example shows how to set up a `sweep` and produce the plots displayed in Fig. 7.4.

The `AWEbox` toolbox was implemented during this thesis together with Rachel Leuthold[1] and Jochem De Schutter[1,2]. It is based in parts on code written by Elena Malz[3] and Mario Zanon[4]. The first version of the `AWEbox` code was implemented by Rachel Leuthold and Jochem De Schutter in 2017. Building on this, the current version of the `AWEbox` was produced during a major refactoring of the previously existing `AWEbox` code in January and February of 2018, done by Rachel Leuthold, Jochem De Schutter and myself. On top of the refactoring, I implemented all features relating to the optimization of nominal and compromised landing trajectories as well as transition trajectories into the `AWEbox`. All of the solutions presented in this thesis are computed with this new version of the `AWEbox`.

[1]Dept. of Microsystems Engineering, University of Freiburg, 79110 Freiburg, Germany.
[2]Kiteswarms GmbH.
[3]Dept. of Electrical Engineering, Chalmers University, 41258 Göteborg, Sweden.
[4]Dynamical Systems, Control and Optimization, IMT Lucca, 55100 Lucca, Italy.

6.2. CasADi and IPOPT

For the implementation of the OCPs we use the C++ software package CasADi [2] that provides a symbolic framework together with automatic differentiation and can be interfaced with python. Once the OCP is formulated in CasADi's symbolic framework, it can be used to build a solver object that automatically generates all necessary derivatives and passes them to the numerical backend. This makes it possible to formulate OCPs in a high level manner with python, while still using fast C++ code to solve the OCPs. The CasADi framework is designed to formulate and solve general NLPs efficiently. This means that one has to first formulate the OCP as a discrete NLP before it can be formulated in CasADi. Thus, discretization schemes like direct collocation, single- or multiple-shooting have to be implemented by the user and are not supplied by CasADi. This requires additional work, but also provides the user with a lot of conceptual freedom regarding the numerical design of the NLP, e.g. how to implement the discretization scheme. At the same time, CasADi provides enough functionality within its symbolic framework to make the implementation of more complex NLPs, which originate from the discretization of an OCP, fast and straightforward. Furthermore, CasADi possesses some helpful structuring and debugging tools that make it easier for the engineer to check whether the desired optimization problem has been correctly translated into code or not.

As numerical backend we choose the interior-point solver IPOPT [41]. Using an interior-point solver can lead to some challenges regarding homotopies. This is due to the fact that interior-point solvers have troubles with warm-starting [44]. We mitigate this by using the homotopy strategy described in section 2.4.

In addition, IPOPT is well documented and supported within the CasADi framework. Even though it is possible to use other solvers within CasADi (e.g. the SQP solver WORHP [7]), IPOPT is the default solver and thus much better documented. Another advantage of IPOPT is that it can exploit the sparsity of a given optimization problem. OCPs usually exhibit some inherent sparsity depending on the way the differential constraints are approximated and discretized. Using the direct collocation method outlined in section 2.2 ensures that each collocation constraint only depends on the variables of its corresponding collocation interval. This translates to a sparse structure of the OCP, resulting in e.g. a sparse Hessian that can be inverted efficiently. To preserve this sparsity, it is important not to introduce dependencies between the decision variables outside of the block structure generated by the collocation intervals, e.g. by using a control parameterization without local support. This would introduce new non-zero entries into the OCP's Hessian and thus destroy the sparsity of the problem. The solver IPOPT, together with CasADi can make full use of the OCP's sparsity patterns to increase its computation speed. Within IPOPT we use linear solvers from the HSL mathematical software library [22].

This chapter served as a brief overview of how the framework discussed in chapter 5 is implemented into the AWEbox. In the following chapters, we will omit further details about the software implementation and focus on a precise mathematical formulation of the OCPs and the analysis of their corresponding solutions.

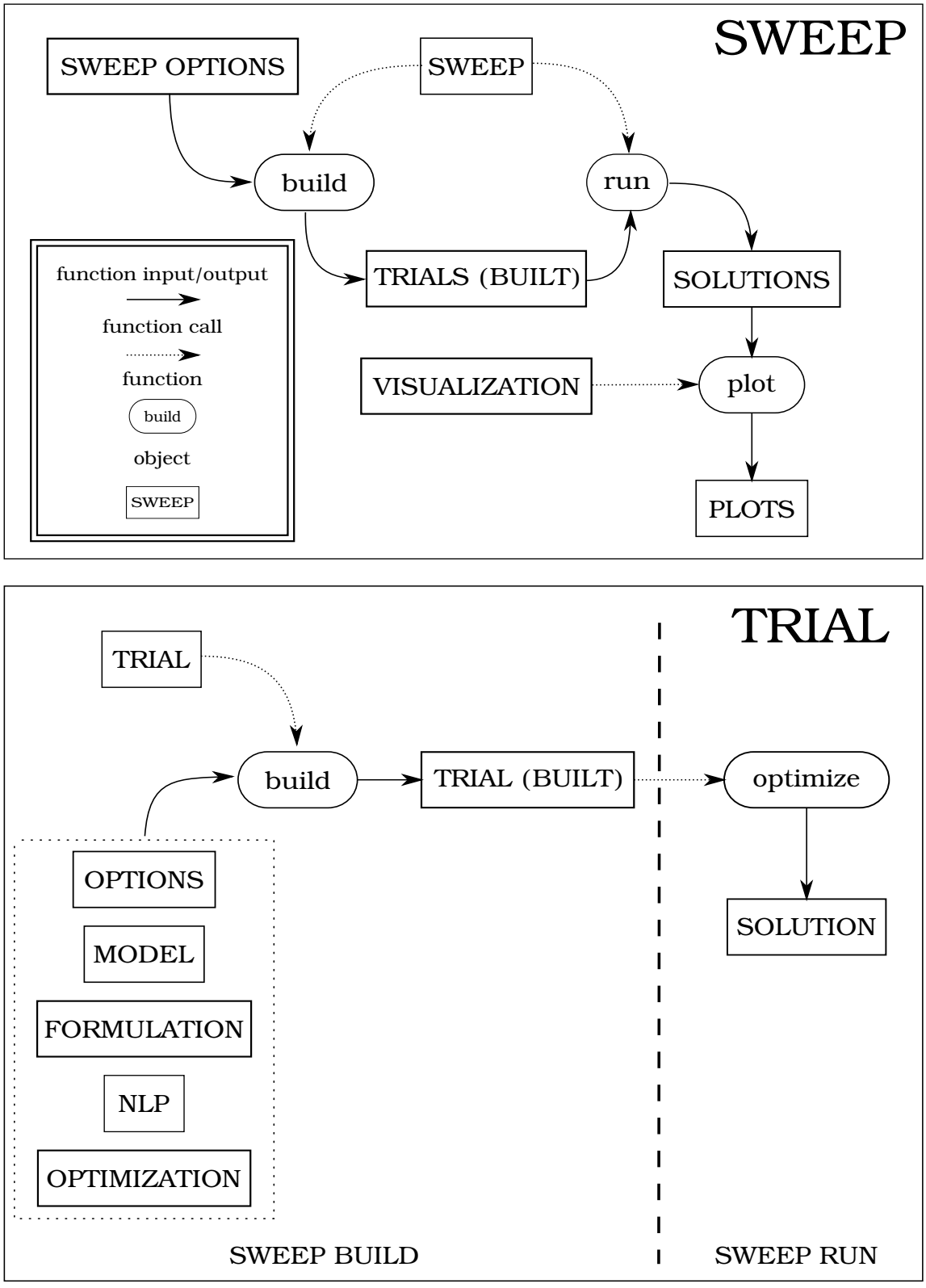

Figure 6.1.: Schematic overview of the `AWEbox` on `sweep` and `trial` level.

7. Landing Trajectories with Nominal Flight Behavior

Now that all the building blocks of the cost function, the constraints and the homotopy schedule have been introduced, we need to explain how these components can be used to find optimal emergency landing trajectories. To do so, this chapter gives a detailed explanation of how optimal nominal landing trajectories, i.e. trajectories that follow emergency response strategy (**B**) from section 3.3, are computed. Emergency response strategy (**B**) implies that the AWES has enough time to continue on the pumping cycle until an optimal exit point is reached. This means that the normalized time parameter ξ_0 of the parameterized initial condition is chosen optimally. It further implies that the AWES exhibits nominal flight behavior during the landing.

The first section will introduce the OCP formulation of the problem as well as the homotopy schedule formalized in the framework that has been laid out in chapter 5. The following section is an in-depth analysis of the solutions of the whole homotopy procedure. This includes a discussion of the solutions of several different homotopy steps, followed by a study of how the optimal solution changes with the reference wind speed. The chapter then concludes with an investigation into how the system's kinetic and potential energy behaves during the course of the trajectory.

7.1. OCP Formulation and Homotopy Schedule

With the preliminary work done in chapter 5, we can formulate the OCP that is used to derive the landing trajectory with nominal flight behavior. The goal of the OCP is to find a trajectory that flies the nodes close to the ground-station while reducing their kinetic energy as much as possible. This will be translated into the OCP with the help of the nominal landing cost and the terminal position inequality that were presented in sections 5.1 and 5.2.2. Just as outlined in section 2.2, we use direct collocation to approximate the continuous OCP.

The discretized OCP formulation is as follows:

$$
\begin{aligned}
\underset{\mathbf{x}, E, \boldsymbol{\lambda}, \mathbf{u}, t_\mathrm{f}, \mathbf{p}, \xi_0}{\text{minimize}} \quad & \eta T + (1-\eta) L + G \\
\text{subject to} \quad & \mathbf{0} = \mathbf{c}_\mathrm{col}(\mathbf{x}_{i,j}, E_{i,j}, \boldsymbol{\lambda}_{i,j}, \mathbf{u}_i, t_\mathrm{f}, \mathbf{p}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_-, \\
& \mathbf{0} = \mathbf{c}_\mathrm{con}(\mathbf{x}_{i+1,0}, E_{i+1,0}, \mathbf{x}_{i,n}, E_{i,n}, \mathbf{p}), && \forall i \in \mathcal{I}^-, \\
& \mathbf{0} = \hat{\mathbf{x}}_0(\xi_0) - \mathbf{x}_{0,0}, \\
& \xi_0 \in [0, 1], \\
& 0 = E_{0,0}, \\
& 0 \leq h_\mathrm{acc}(\ddot{\mathbf{q}}_{a,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, a \in \mathcal{A}, \\
& 0 \leq h_\mathrm{ts}(\lambda_{k,i,j}, l_{k,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_-, k \in \mathcal{T}, \\
& 0 \leq h_\mathrm{acol}(\mathbf{q}_{a_1,i,j}, \mathbf{q}_{a_2,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, (a_1, a_2) \in \mathcal{A}_\mathrm{com}, \\
& 0 \leq h_\mathrm{tp}(\mathbf{q}_{p,\bar{m},n}), && \forall p \in \mathcal{N}, \\
& \mathbf{0} \leq \boldsymbol{\lambda}, \\
& \mathbf{u}_\mathrm{min} \leq \mathbf{u} \leq \mathbf{u}_\mathrm{max}, \\
& \mathbf{x}_\mathrm{min} \leq \mathbf{x} \leq \mathbf{x}_\mathrm{max}, \\
& \mathbf{p}_\mathrm{min} \leq \mathbf{p} \leq \mathbf{p}_\mathrm{max}, \\
& 0 \leq t_\mathrm{f}.
\end{aligned}
\tag{7.1}
$$

The optimization variables for (7.1) comprise the system's state $\mathbf{x}$, the energy E, the algebraic variables $\boldsymbol{\lambda}$, the controls $\mathbf{u}$, the final time t_f as well as the homotopy parameters $\mathbf{p}$ and the normalized time parameter ξ_0. The cost function is a combination of the tracking cost T, landing cost L and general problem cost G. The equality constraints are made up of collocation, continuity and initial energy constraints. The inequality constraints of the OCP are the maximum acceleration constraint, the maximum tether stress constraint, the anti-collision constraint, the terminal position constraint as well as bounds on $\boldsymbol{\lambda}$, $\mathbf{u}$, $\mathbf{x}$, $\mathbf{p}$ and ξ_0. Note that this formulation represents all the homotopy steps that are performed to solve the problem. Hence, the homotopy parameters $\mathbf{p}$ are included in the optimization variables. For each homotopy step, the objective, dynamics and constraints of (7.1) are changed as explained in sections 2.4 and 5.3. The exact homotopy schedule is presented in table 7.1.

We set the number of collocation intervals to $m = 50$ and the degree of the Lagrange polynomials to $d = 4$. For a dual-kite system, we have 25 states (including E), 3 algebraic variables, 5 controls and 10 parameters (including two bounded homotopy parameters, $\boldsymbol{\theta} = \mathbf{F}_\mathrm{F} \in \mathbb{R}^6$ as well as t_f and ξ_0). Since there is one control vector per collocation interval, one state vector per control point and one algebraic variable vector per collocation point, (7.1) has 7110 decision variables. Furthermore, there are 6801 equality and 4253 inequality constraints.

Table 7.1.: Nominal landing homotopy schedule.

homotopy step	initial	tracking	nom. landing	tether	final
objective	×	×	$T \xrightarrow{\eta} L$	×	×
dynamics	×	$\mathbf{F}_{\mathrm{FA}} \xrightarrow[\mathbf{F}_{\mathrm{F}}]{\gamma} \mathbf{F}_{\mathrm{A}}$	×	$\mathbf{F}^{\mathrm{t}}_{\mathrm{TD}} \xrightarrow{\tau} \mathbf{F}^{\mathrm{e}}_{\mathrm{TD}}$	×
constraints	×	×	×	×	×

7.2. Analysis of Solutions

After defining the OCP formulation and homotopy schedule for the landing trajectory with nominal fight behavior, we need to discuss the resulting solutions. To do so, we will evaluate the solutions of several different homotopy steps, give an analysis of the optimal solution's sensitivity to a change in the reference wind speed, as well as a study of how the kinetic and potential energy of the system behaves during the landing.

7.2.1. Homotopy Steps

To better comprehend how the solution evolves during the homotopy procedure, we discuss a number of intermediate solutions. With this, we can get an understanding of how the solution changes with each deformation of the cost function, dynamics or constraints.

Initial Homotopy Step

The initial homotopy step is used to track the initial guess using the fictitious forces $\mathbf{F}_{\mathrm{F}}$ together with the true aerodynamic forces $\mathbf{F}_{\mathrm{A}}$. Since $\mathbf{F}_{\mathrm{F}}$ is not bounded or penalized in the cost function, the result of this homotopy step should be the initial guess itself.

The resulting trajectory is shown in Fig. 7.1, with the kite at node 1 and 2 presented in red and green respectively. The pumping cycle used as parameterized initial condition is displayed in black. The solution is close to the initial guess, but clearly not identical, especially due to the small inward curve at the beginning of the trajectory. This behavior has four reasons. Firstly, there is a small regularization on the fictitious forces and node accelerations in the general problem cost G. Hence, the solution depicted in Fig. 7.1 can actually have a lower cost than the initial guess itself. Secondly, we are not tracking exactly the initial guess, but add a very small stagger distance to increase convergence as described in section 5.5.4. Thirdly, the solvers accuracy in the initial homotopy step is set at $\tau_{\mathrm{f}} = \tau_{\mathrm{b}}$ and $r_{\mathrm{tol}} = r_{\mathrm{a}}$, meaning that the OCP is not solved for full accuracy. Last but not least, the parameterized initial constraint only holds in parts for the initial guess. Most prominently, it does not hold for the node velocities $\dot{\mathbf{q}}$. This means that the initial

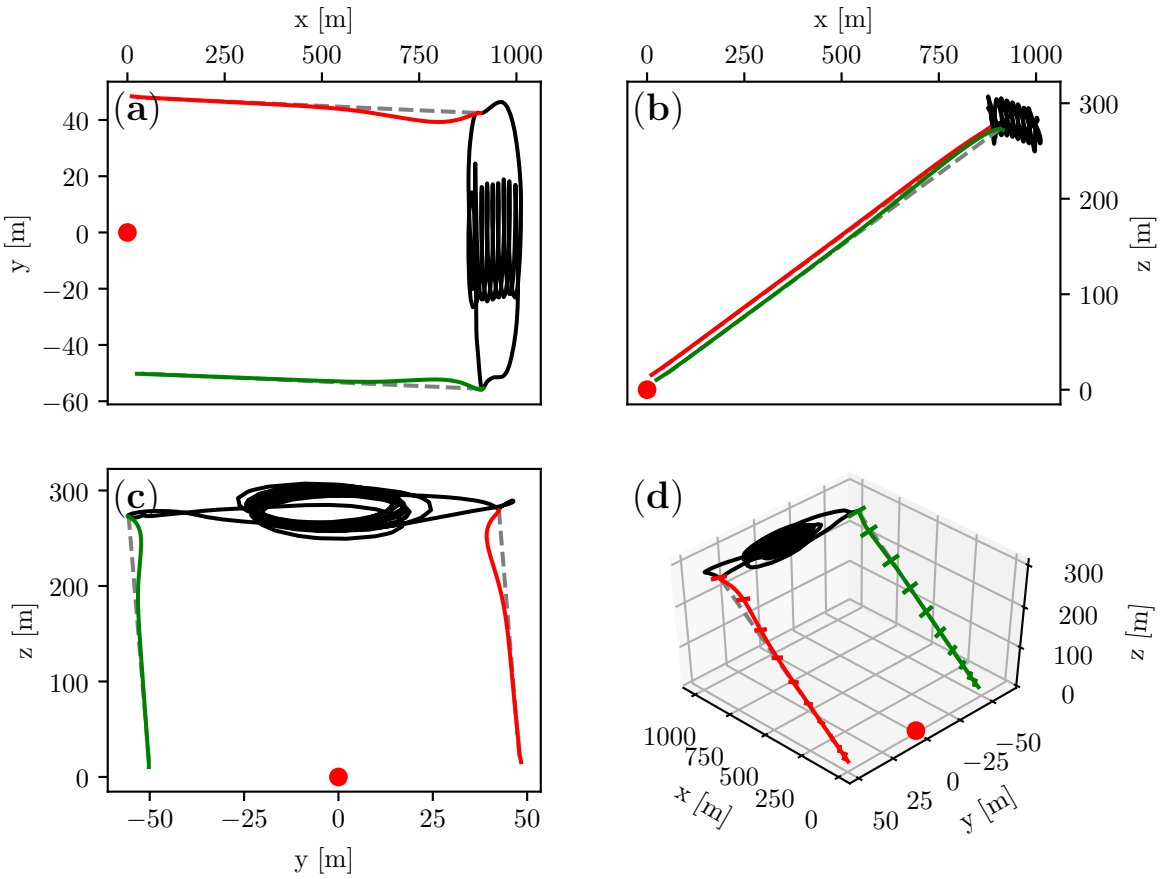

Figure 7.1.: Nominal landing trajectory. Solution after initial homotopy step compared to initial guess (dashed). The initial trajectory is displayed in black. Position of the ground-station represented by a red dot.

guess cannot be tracked since it is not feasible.

Tracking Homotopy Step

In the tracking homotopy step, the fictitious forces are decreased to zero and exchanged for the true aerodynamic forces. Hence, the solution should result in the closest (meaning the smallest tracking cost) approximation of the tracking reference that follows the true system dynamics without fictitious forces. Indeed, Fig. 7.2 shows that the solution to the tracking homotopy step still resembles the initial guess, but not as closely as after the initial homotopy step due to the lack of fictitious forces. The fact that the real system is able to track the initial guess reasonably well is confirmation that the choice of initial guess is sufficient. The homotopy strategy has successfully produced a feasible initial guess.

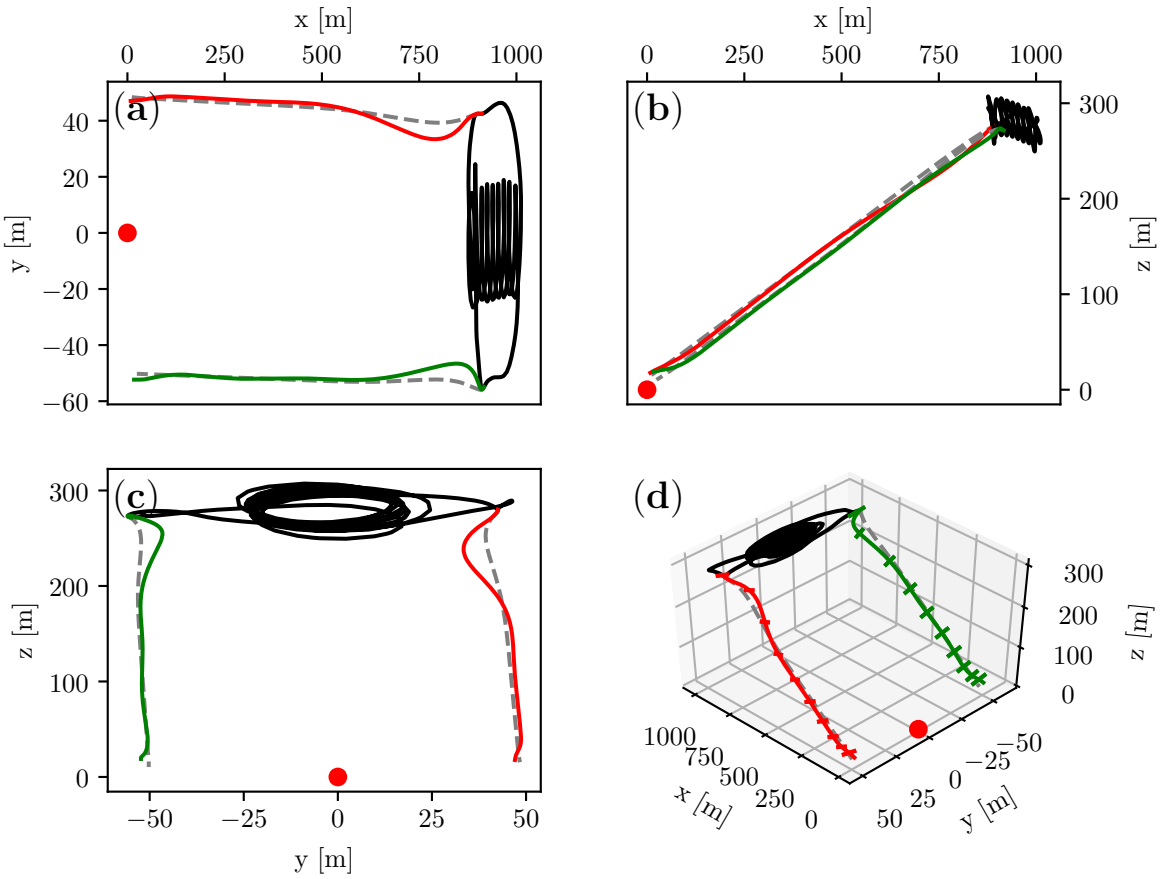

Figure 7.2.: Nominal landing trajectory. Solution after tracking homotopy step compared to solution of previous homotopy step (dashed). Initial trajectory displayed in black. Position of the ground-station represented by a red dot.

Nominal Landing Homotopy Step

Now that we have a feasible initial guess, the nominal landing homotopy step changes the cost function from a tracking problem to the nominal landing problem that we actually want to solve. Since the new cost function penalizes the terminal velocity of all nodes, we expect to find a breaking maneuver at the end of the solution of this homotopy step. Due to the non-convexity of the problem and the reasonable choice and feasibility of the initial guess produced in the last homotopy steps, the basic shape of the solution displayed in Fig. 7.3 is expected to be roughly the same as before.

The solution indeed resembles the initial guess, but for an upwards breaking-motion at the end. Both kites turn inward and generate additional lift by increasing the lift coefficient $C_{\mathrm{L},a}$. This has two effects: Firstly, the projections of the kites' velocities on the negative z-axis are increasing. This means that the kites are transforming kinetic

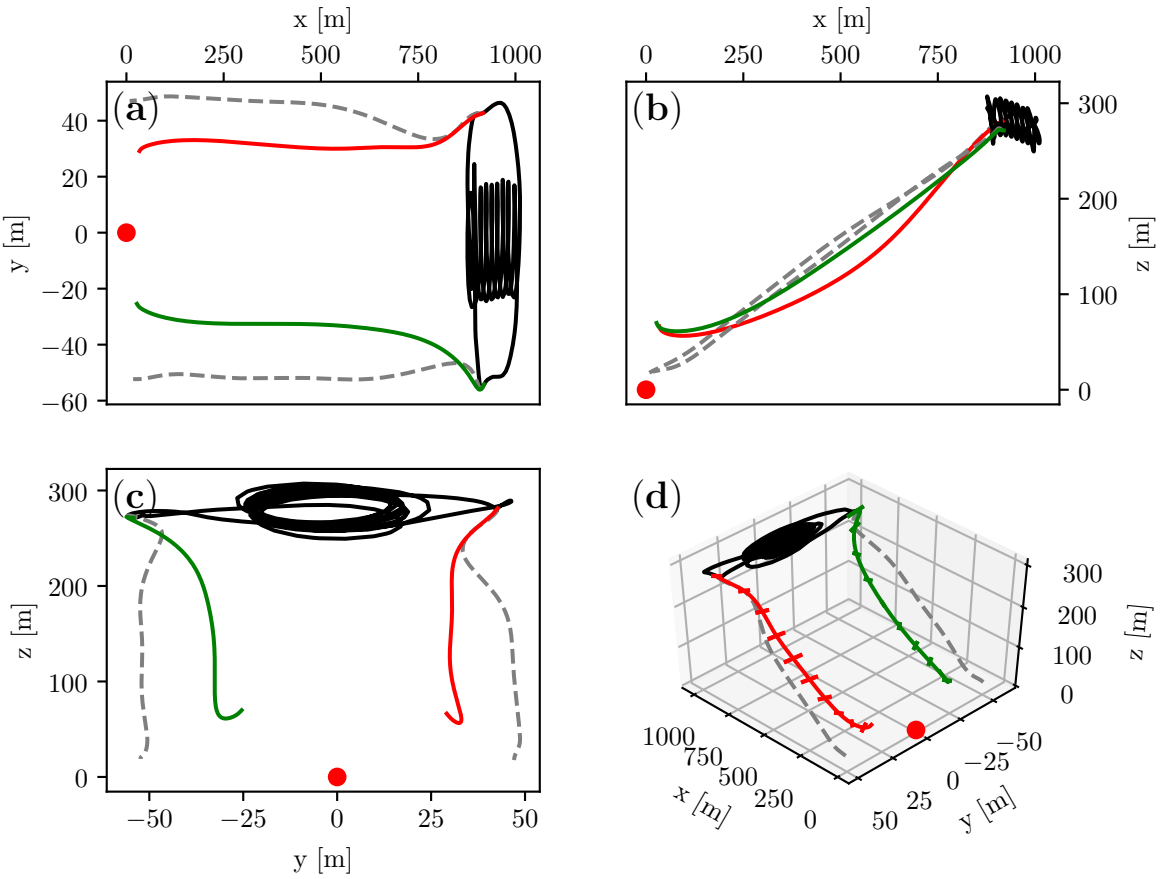

Figure 7.3.: Nominal landing trajectory. Solution after nominal landing homotopy step compared to solution of previous homotopy step (dashed). Initial trajectory displayed in black. Position of the ground-station represented by a red dot.

energy into potential energy, thereby slowing themselves down. Secondly, the increase in $C_{\mathrm{L},a}$ also leads to an increase in the drag coefficient $C_{\mathrm{D},a}$ since the induced drag of a kite a is quadratic in $C_{\mathrm{L},a}$ (cf. (4.20)). An increase in $C_{\mathrm{D},a}$ in turn increases the drag force $\mathbf{F}_{\mathrm{D},a}$ that is linear in $C_{\mathrm{D},a}$ and points parallel to the kite's relative velocity (cf. (4.22)), thus decelerating it. The extent of the breaking maneuver is limited by the active terminal position inequality constraint that prevents the kites from increasing their height more. The middle node $\mathbf{q}_1$ is consequently decelerated through the consistency constraints. Since this homotopy step is using a trivial tether drag model, the tethers' orientation to their respective relative velocities is immaterial to the amount of drag they produce, and hence for the shape of the trajectory.

Tether Homotopy Step

The tether drag homotopy step is used to switch from the trivial tether drag model to a more sophisticated model. To analyze the influence of the tether drag model on the optimal solution, we compare the solutions of this homotopy step generated by the three tether drag models that have been introduced in section 4.4.2. Namely these are, in growing order of sophistication, the trivial, the projected and the equivalence model. Fig. 7.4 shows a comparison of the solutions generated by the three models.

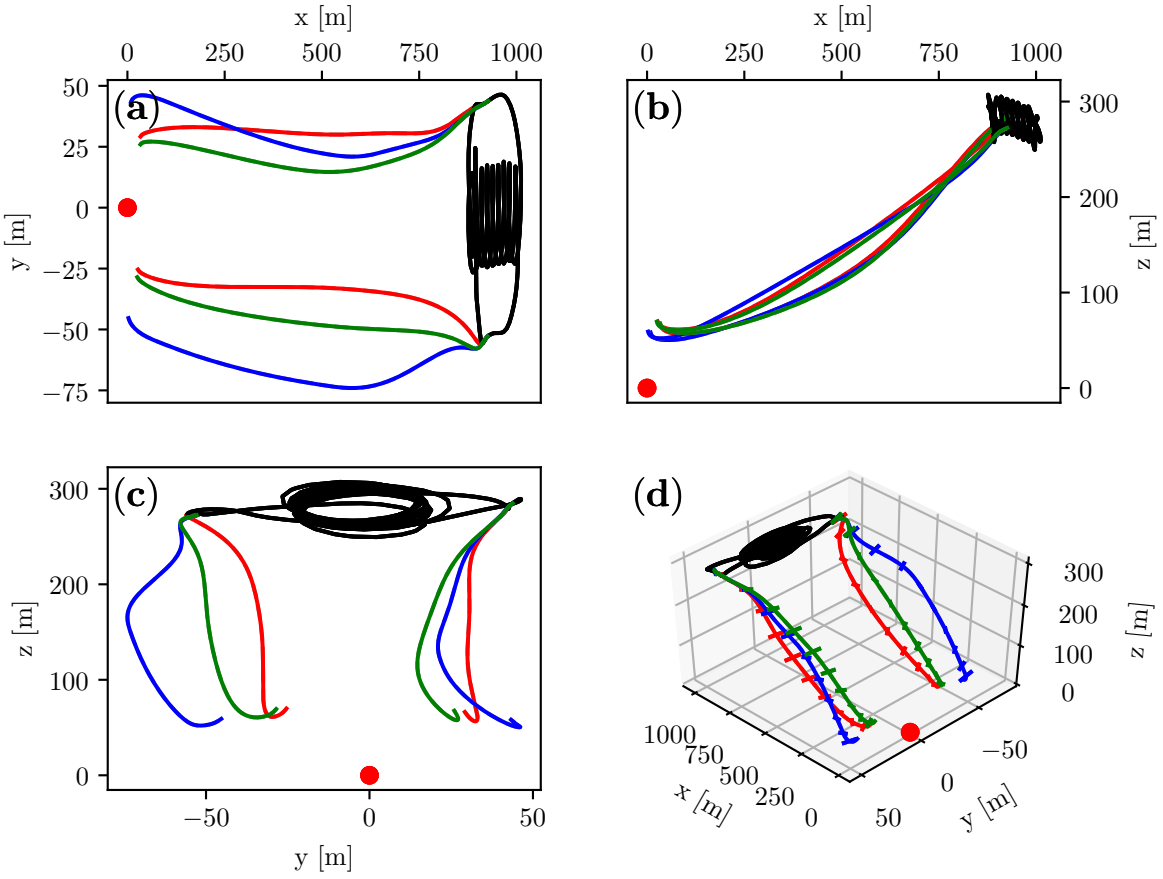

Figure 7.4.: Nominal landing trajectory. Comparison of trivial (red), projected (blue) and equivalence (green) tether drag model. Initial trajectory displayed in black. Position of the ground-station represented by a red dot.

The different trajectories show that the choice of tether drag model is significant for the shape of the solution. The solution using the trivial tether drag model (red) is identical to the solution of the previous homotopy step. The solution of the projected tether drag model (blue) is characterized by a larger distance of the two kites. They are around 100 m apart at all times. With a secondary tether length of 50 m, this means

that the two secondary tethers approximately form a line connecting the two kites with $\mathbf{q}_1$ in the middle. This makes sense because it increases $l_{\perp,k}$, the length of the secondary tethers that is perpendicular to its relative velocity (cf. (4.26)).

In Fig. 7.4 (a) one can also see that the trajectory corresponding to the projected tether drag model is curved. Since both the problem formulation and the initial guess produced in the previous homotopy step are symmetrical with respect to the x-axis, this must be due to the asymmetry in the pumping trajectory that is used as parameterized initial condition. Indeed we can see in Fig. 7.4 (a) that the kites continue the curvature of the pumping cycle. To change this curvature would mean to abandon the perpendicular orientation of the secondary tethers for some time and is therefore not optimal.

The solution generated with the equivalence model (green) does not exhibit the same large distance between the kites, even though the model does consider the tethers' orientations to their respective relative velocities. This can be explained as follows: The equivalence model produces more tether drag than the projected model. Hence, it does not need to adjust the tethers' orientation in order to reduce the terminal node velocity below 1 $^{\mathrm{m}}/_{\mathrm{s}}$. At this point, the nominal landing cost L is so small that it does no longer dominate the cost function, but the regularization cost R does instead. Thus, the tethers' orientation is not changed even though it would produce additional tether drag and decelerate the kites further. One can argue that it is not desirable for a regularization term to affect the optimal solution in this way and that therefore the tuning parameters of the objective function should be changed. We have however come to the conclusion that a "smoother" trajectory is more desirable than a minuscule decrease in terminal velocity, considering that the velocity is already below 1 $^{\mathrm{m}}/_{\mathrm{s}}$. The fact that the regularization cost is larger than the nominal landing cost speaks for the fact that the AWES can easily reduce its terminal energy.

In order to get a better understanding of the different tether drag models, we define the accelerations

$$a_{\mathcal{A}} = \sum_{a \in \mathcal{A}} \frac{1}{m_a} F^{\mathcal{N}}_{\mathrm{TD},a} \tag{7.2}$$

$$a_1 = \frac{1}{m_1} F^{\mathcal{N}}_{\mathrm{TD},1} \tag{7.3}$$

$$a_{\mathcal{N}} = \sum_{p \in \mathcal{N}} \frac{1}{m_p} F^{\mathcal{N}}_{\mathrm{TD},p}. \tag{7.4}$$

The first acceleration $a_{\mathcal{A}}$ denotes the sum of accelerations on all kite nodes $a \in \mathcal{A}$ that result from tether drag forces. The second acceleration a_1 results from the tether drag forces on the center node $p = 1$. The third acceleration $a_{\mathcal{N}}$ is the sum of $a_{\mathcal{A}}$ and a_1, so it is referring to all nodes $p \in \mathcal{N}$. Fig. 7.5 shows a comparison of $a_{\mathcal{A}}$, a_1 and $a_{\mathcal{N}}$ for the three tether drag models.

The acceleration $a_{\mathcal{A}}$ is in the same order of magnitude for all tether drag models. This confirms the argument presented above: Even though the solution of the equivalence

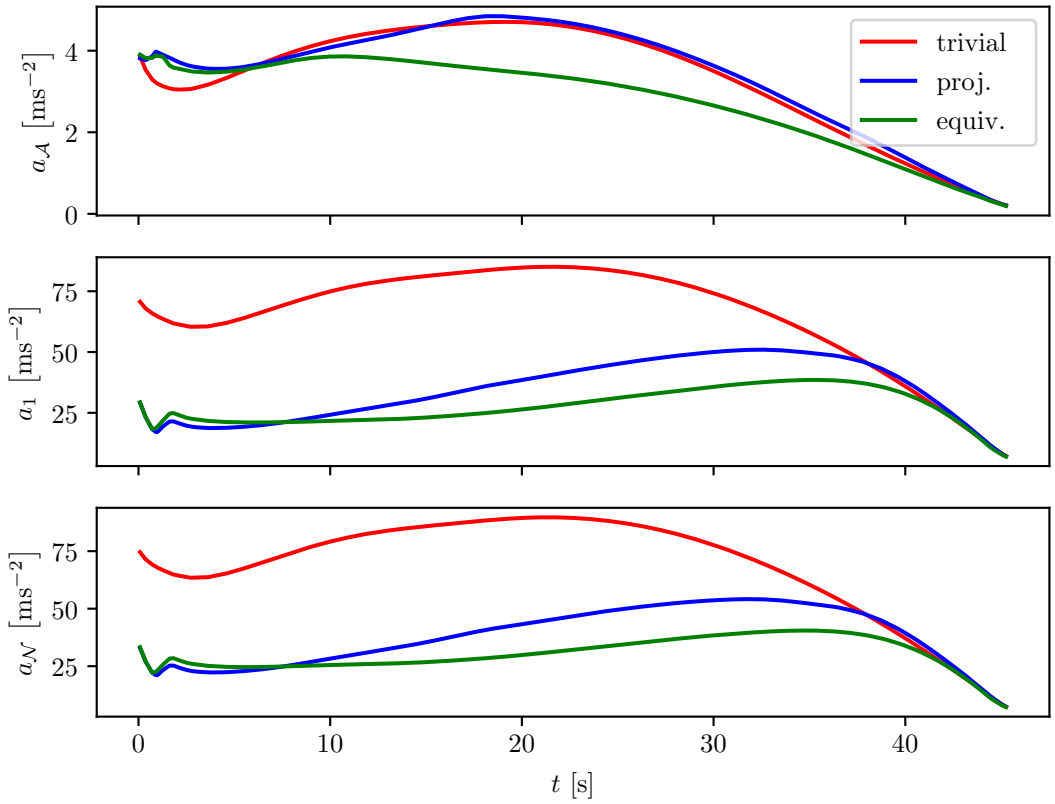

Figure 7.5.: Nominal landing trajectory. Comparison of the accelerations $a_{\mathcal{A}}$, a_1 and $a_{\mathcal{N}}$ for the trivial, projected and equivalence tether drag models.

tether drag model does not adapt its tether orientations to increase $l_{\mathrm{k},\perp}$ like the projected tether drag model, the tether drag forces acting on the kites are roughly the same. When looking at the plot for a_1, we see that the projected and equivalence model look again similar, however the trivial model exhibits much higher tether drag accelerations on the center node $p = 1$. This aligns with the equations describing the trivial tether drag model. Equation (4.22) shows that the trivial tether drag grows linearly with the (unprojected) tether length l_k and quadratically with the tether's relative velocity $\mathbf{u}_{\mathrm{T},k}$. Since $l_{\mathrm{T}} \gg l_{\mathrm{T},\perp}$ for most of the trajectory, there is an overestimation of a_1 within the trivial tether drag model. Hence, we can argue that the trivial tether drag model is not suitable for the computation of landing trajectories.

Even though the equivalence tether drag model is a better approximation of the real system than the projected and trivial model, using this more sophisticated model comes at a price. Fig. 7.6 shows a comparison of the tether drag models with respect to four nu-

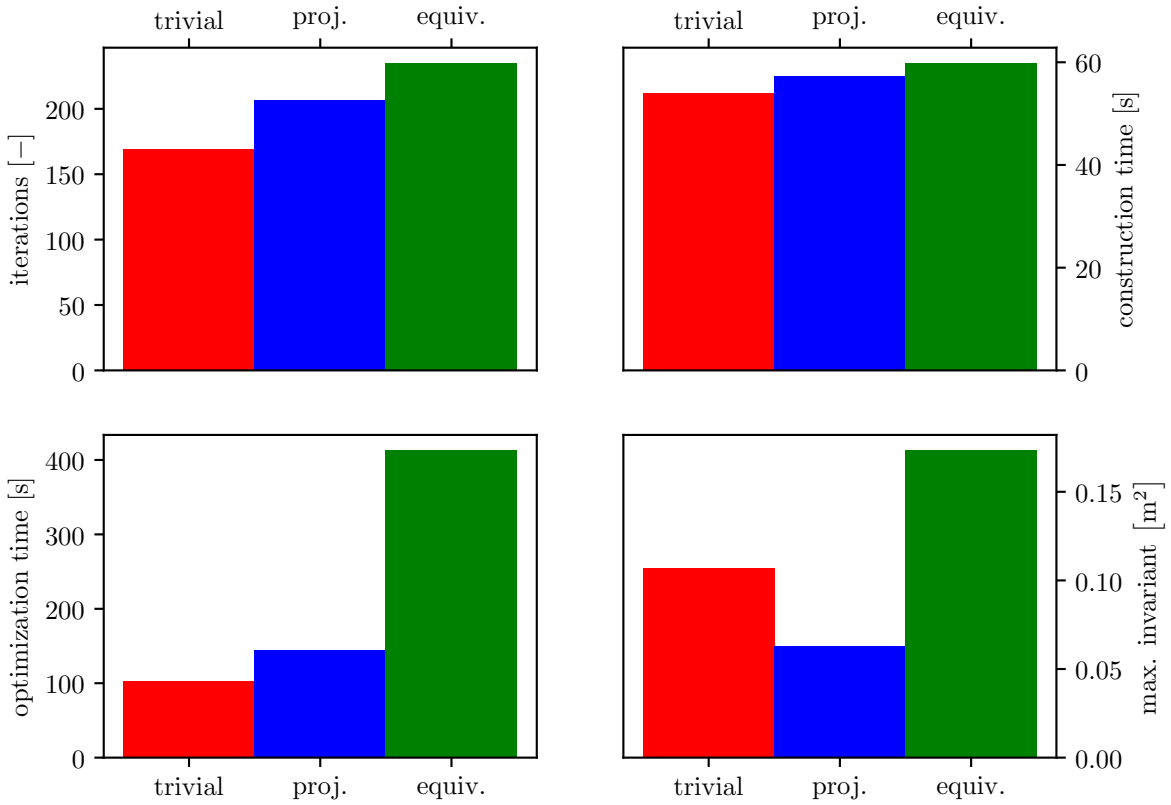

Figure 7.6.: Nominal landing trajectory. Comparison of the number of iterations, NLP construction time, optimization time and maximum invariant for the trivial, projected and equivalence tether drag models.

merical parameters. Specifically, these parameters are the number of iterations needed to solve the optimization problem (in total, so including all homotopy steps), the construction time of the NLP that is passed to the numerical solver, the optimization time needed by the numerical solver and the maximum invariant of the corresponding solution. The latter is a useful criterion when judging whether a solution is indeed physical or whether it exhibits some unphysical behavior made possible by e.g. discretization errors.

The number of solver iterations increases by 22 % when changing from the trivial to the projected tether drag model, then increases again by 14 % when changing to the equivalence tether drag model. Regarding the NLP construction time, one can only observe negligible increases when compared to the optimization time. Here, we see a jump of 40 % from the trivial to the projected model and then again an increase in 187 %, almost tripling the optimization time. The absolute computation times are less

relevant, since they depend on the system that is used for computation.

Regarding the maximum invariant, the projected model performs best, followed by the trivial and then the equivalence model. The difference here is however not relevant since even the maximum invariant of the equivalence model is at 0.17 m^2, which corresponds to an error of 0.8 % relative to the secondary tether length of 50 m. Hence, all three solutions can be considered physical.

Final Homotopy Step

The final homotopy step is used to solve the same problem as at the end of the previous homotopy step, but with higher accuracy. Explicitly, this means setting the target barrier parameter $\tau_f = \tau_c$. The initial barrier parameter $\tau_0 = \tau_b$ stays the same as before. Additionally, the tolerance is set to $r_{tol} = r_b$. The final homotopy step is performed using the equivalence tether drag model. As a result, we expect the solution of the previous homotopy step with small changes due to the additional accuracy.

The solution shown in Fig. 7.7 is indeed almost identical to the previous solution. This solution represents the last homotopy step and is therefore the solution to the problem we initially set out to find. It is also the only one that is computed with full accuracy and therefore the only solution that can be considered as truly representing the system dynamics and constraints. All intermediate solutions are interesting as numerical artifacts that help understand and analyze how the solution to the final homotopy step is created, but there is no guarantee that they are optimal or feasible with respect to the final problem formulation and accuracy. The optimal landing trajectory computed in the final homotopy step takes approximately 45 s.

7.2.2. Sensitivity to Wind Speed

Now that we have analyzed how the solution changes over the course of the homotopy procedure and how sensitive it is to the choice of the tether drag model, we need to investigate how sensitive the solution is to parameter changes. The most relevant of these parameters is the reference wind speed u_{ref} because it has been found to be a crucial factor in determining the optimal shape of dual-kite pumping cycles [24]. To analyze the sensitivity of the landing trajectories to a change in u_{ref}, its value is varied by $\pm$ 50 % around its default value of 5 $^m/_s$, with a step size of 0.5 $^m/_s$. The results of this analysis can be seen in Fig. 7.8, comparing the average terminal velocity $\dot{q}_{\bar{m},n,avg}$ of all nodes, maximum and average tension τ_{max} and τ_{avg} in the main tether as well as ξ_0, the optimal starting point on the parameterized initial condition.

The terminal velocity $\dot{q}_{\bar{m},n,avg}$ gets smaller with a growing u_{ref}. This makes sense since the kite and tether drag forces all grow quadratically with the relative wind speed. Since the landing is performed anti-parallel to the wind, this means that for a larger u_{ref}, the drag forces also increase, slowing the nodes down further. This however is only true until the nodes terminal velocity becomes so small that the landing cost L is no longer the

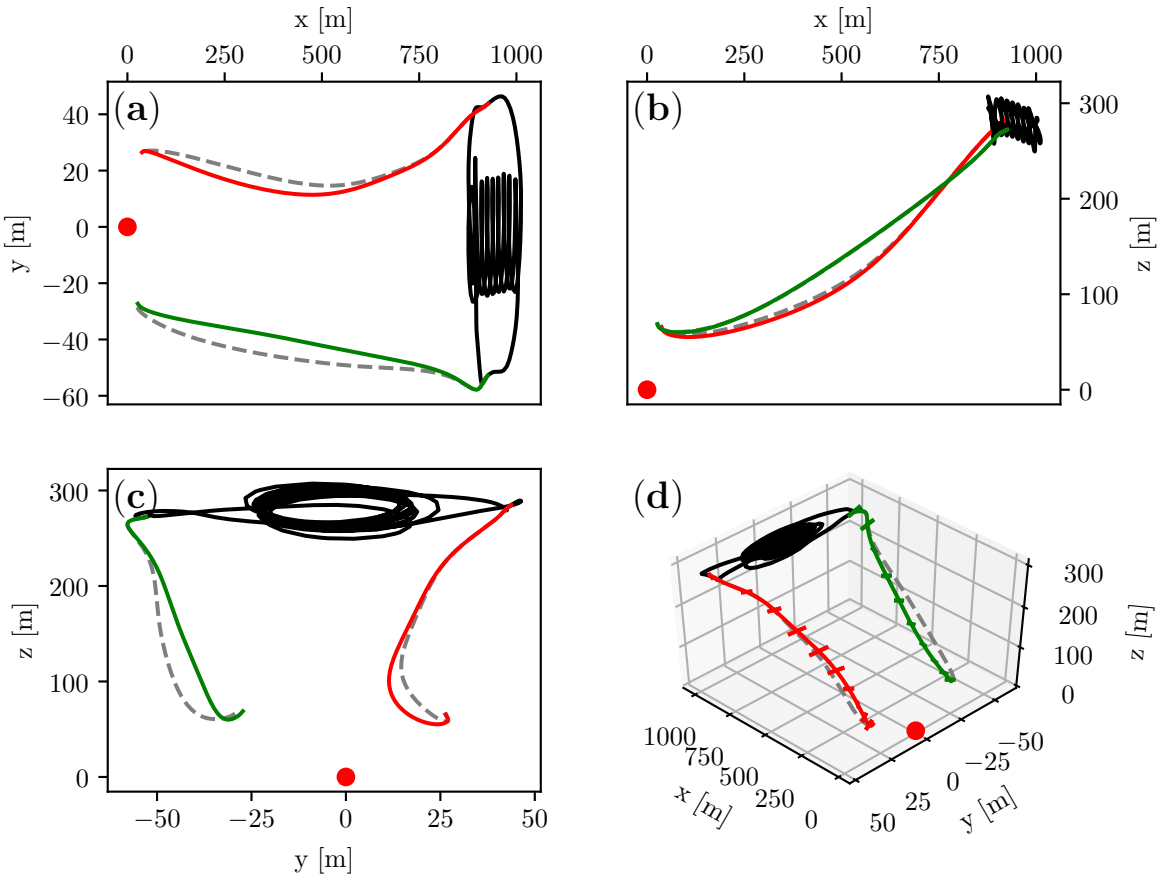

Figure 7.7.: Nominal landing trajectory. Solution after final homotopy step with equivalence tether drag model compared to solution of previous homotopy step (dashed). Initial trajectory displayed in black. Position of the ground-station represented by a red dot.

dominant term of the cost function. At this point, there can be small increases in $\dot{q}_{\bar{m},n,\text{avg}}$ if it in turn reduces the regularization cost R. Hence, the cost function must be tuned in such a way that R is only the dominant term of the cost function if the terminal node velocities and thus L are small enough.

With a growing u_{ref}, the average main tether tension τ_{avg} is also growing. Increasing the wind speed results in an increase in the kites' drag forces. Since the kites' velocity is partly in the direction of the main tether, a part of the force is compensated by tether tension in the main tether. One can see that from 2.5 m/s to 3.0 m/s, τ_{avg} does not increase. This can be explained when looking at the values for the maximum main tether tension τ_{max}. Here the relationship to a change in u_{ref} is not as straightforward. Indeed, the largest value of $\tau_{\text{max}} \approx 535$ N corresponds to the smallest $u_{\text{ref}} = 2.5$ m/s, hence the

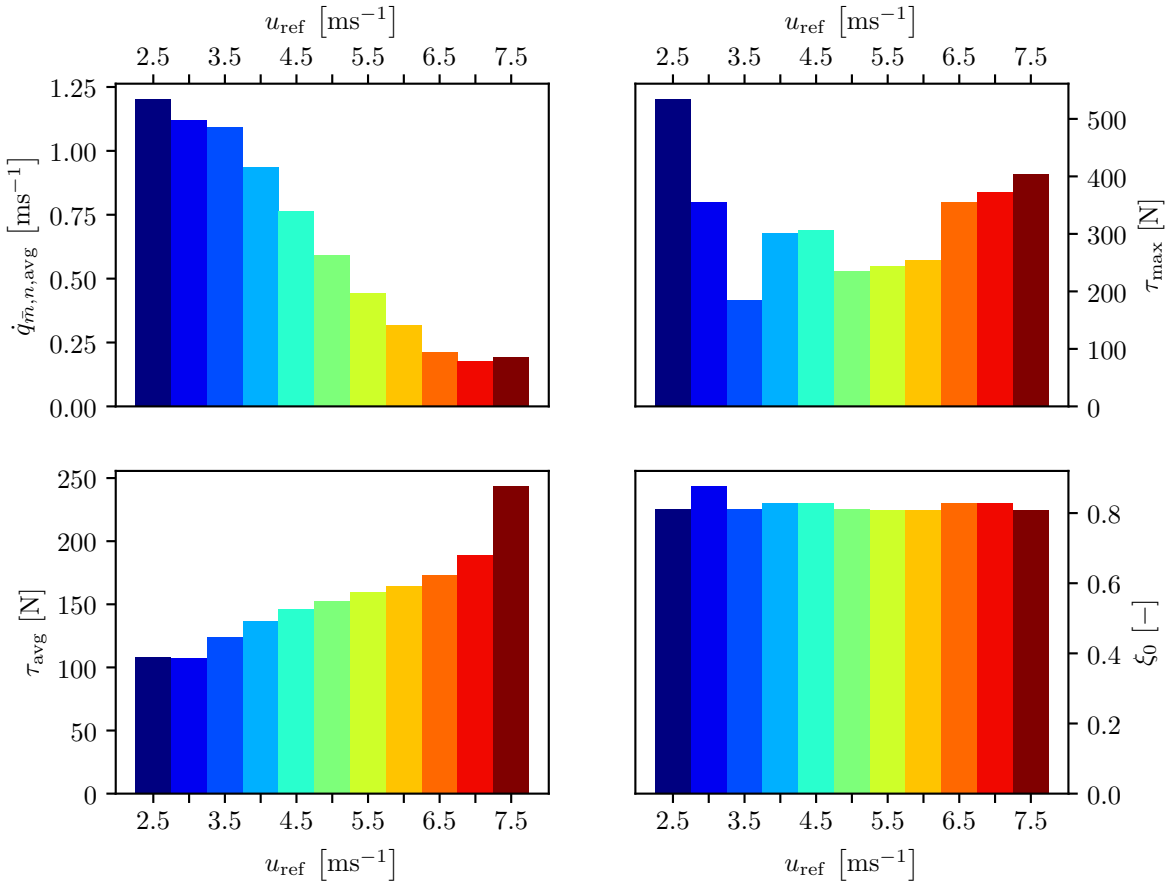

Figure 7.8.: Nominal landing trajectory. Sensitivity to change in u_{ref}. Comparison of average terminal node velocity, maximum and average tether tension and optimal initial trajectory parameter.

increase of τ_{avg} from 2.5 m/s to 3.0 m/s. The largest tether tension usually occurs during the breaking maneuver at the end of the trajectory. Decreasing the wind speed generally leads to a smaller main tether tension, but can also result in large values for τ_{max} when u_{ref} becomes too small and the kites have to perform more aggressive breaking maneuvers in order to decrease their terminal velocity. A value of $\tau_{\text{max}} \approx 535$ N is however far from the values observed for pumping cycles (7 kN for $u_{\text{ref}} = 5.0$ m/s).

In Fig. 7.8 we can also see that the optimal starting point on the parameterized trajectory only varies by + 5/− 2 %. This means that ξ_0 is robust to a change in u_{ref} and that the constraint $\xi_0 \in [0, 1]$ never becomes active.

7.2.3. Potential and Kinetic Energy

An analysis of the AWES's energy flow can give further insight into the optimal landing trajectory. At the end of the landing trajectory, the height of each node must decrease such that the solution is feasible with respect to the terminal position constraint. Also, the landing cost penalizes high terminal node velocities. Hence, the system must both lose kinetic and potential energy during the landing. This process is shown in Fig. 7.9, both for each individual node as well as for the whole system.

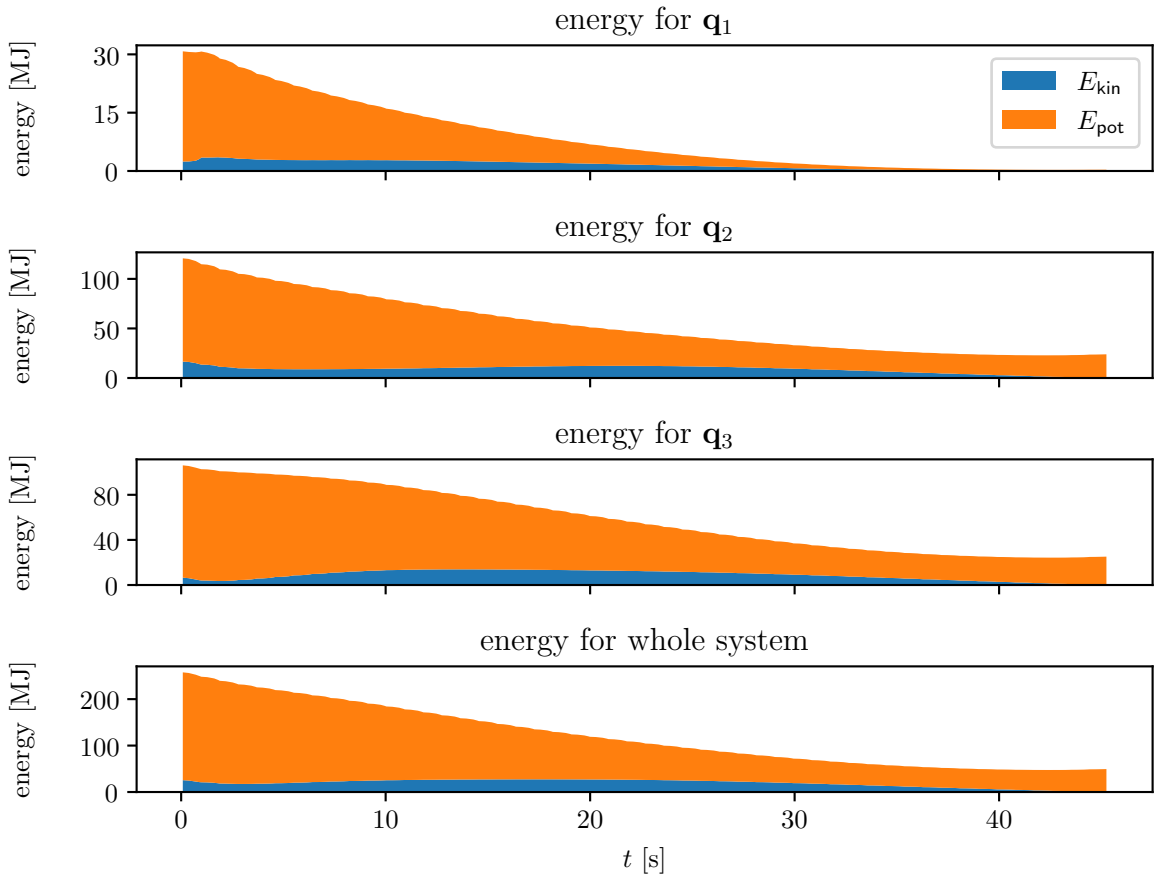

Figure 7.9.: Nominal landing trajectory. Kinetic and potential energy over time for individual nodes and whole system.

The overall energy of the system is steadily decreasing. At first, potential energy is converted into kinetic energy, thus reducing the height of the nodes but increasing their velocities. The increase in the node velocities causes an increase in the drag forces that reduce the kinetic energy of the system by emitting energy from it. Thus, the potential energy of each node decreases more than the kinetic energy increases. During the last part of the trajectory, there is a slight increase in the kites' potential energy due to the

upwards breaking maneuver (cf. Fig. 7.7 (b)). The difference between the kites' smallest height and their terminal height is approximately 10 m for both kites. In the same time interval, the breaking maneuver decelerates the kites from 14 $^{\mathrm{m}}/_{\mathrm{s}}$ to 0.8 $^{\mathrm{m}}/_{\mathrm{s}}$. With the exact values for all nodes, this amounts to an increase of 1.9 MJ to the whole system, which is only 0.7 % of the initial energy of 158 MJ. The overall energy of the system is reduced to 19 % of its initial value. The potential energy is reduced to 21 %, while the kinetic energy is reduced to 0.3 %. The kite nodes carry about three times more energy than the middle node $\mathbf{q}_1$ due to the additional weight of the kites, the larger initial height and the larger initial velocity of those nodes. The small oscillation in the potential energy of all nodes is caused by the dynamics of the Baumgarte stabilization, which directly affects the nodes' positions $\mathbf{q}$ and therefore also the potential energy of each node.

In this chapter, we gave an in depth analysis of nominal landing trajectories that are used in accordance with emergency response strategy (**B**). After stating the OCP formulation and the homotopy schedule, we discussed how the optimal solution changes throughout the homotopy scheme. We showed that the choice of tether drag model is significant for the shape of the optimal solution and that it is possible to find landing trajectories with small terminal velocities also for small reference wind speeds. We have also analyzed the energy flow of the system to better understand how the AWES reduces its kinetic energy throughout the landing trajectory. Now that we have given a detailed study of the nominal landing trajectory, we will dedicate the next chapter to giving an overview of all the other emergency trajectories that have been developed during the course of this thesis.

8. Overview of Compromised Landing and Transition Trajectories

This chapter will give an overview of all the landing trajectories that were implemented during the course of this thesis besides the nominal landing trajectory discussed in chapter 7. Specifically, these are transition trajectories (cf. section 8.1) and compromised landing trajectories (cf. section 8.2). The latter cover the emergency scenarios of compromised actuation, a compromised on-board battery and structural damages. Due to time constraints, it is not possible to present an analysis of these trajectories with the same degree of detail as shown in chapter 7.

8.1. Transition Trajectories

Emergency response strategy (**A**) from section 3.3 stipulates the transition of the AWES from one pumping trajectory to another one that is located closer to the ground, all with nominal flight behavior of the AWES. This problem can be formulated with the following OCP:

$$
\begin{aligned}
\underset{\mathbf{x}, E, \boldsymbol{\lambda}, \mathbf{u}, t_\mathrm{f}, \mathbf{p}, \xi_0, \xi_\mathrm{f}}{\text{minimize}} \quad & v T + (1-v) K + G \\
\text{subject to} \quad & \mathbf{0} = \mathbf{c}_\mathrm{col}(\mathbf{x}_{i,j}, E_{i,j}, \boldsymbol{\lambda}_{i,j}, \mathbf{u}_i, t_\mathrm{f}, \mathbf{p}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_-, \\
& \mathbf{0} = \mathbf{c}_\mathrm{con}(\mathbf{x}_{i+1,0}, E_{i+1,0}, \mathbf{x}_{i,n}, E_{i,n}, \mathbf{p}), && \forall i \in \mathcal{I}^-, \\
& \mathbf{0} = \hat{\mathbf{x}}_0(\xi_0) - \mathbf{x}_{0,0}, \\
& \mathbf{0} = \hat{\mathbf{x}}_\mathrm{f}(\xi_\mathrm{f}) - \mathbf{x}_{\bar{m},n}, \\
& [\xi_0, \xi_\mathrm{f}] \in [0,1] \times [0,1], \\
& 0 = E_{0,0}, \\
& 0 \leq h_\mathrm{acc}(\ddot{\mathbf{q}}_{a,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, a \in \mathcal{A}, \\
& 0 \leq h_\mathrm{ts}(\lambda_{k,i,j}, l_{k,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_-, k \in \mathcal{T}, \\
& 0 \leq h_\mathrm{acol}(\mathbf{q}_{a_1,i,j}, \mathbf{q}_{a_2,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, (a_1, a_2) \in \mathcal{A}_\mathrm{com}, \\
& \mathbf{0} \leq \boldsymbol{\lambda}, \\
& \mathbf{u}_\mathrm{min} \leq \mathbf{u} \leq \mathbf{u}_\mathrm{max}, \\
& \mathbf{x}_\mathrm{min} \leq \mathbf{x} \leq \mathbf{x}_\mathrm{max}, \\
& \mathbf{p}_\mathrm{min} \leq \mathbf{p} \leq \mathbf{p}_\mathrm{max}, \\
& 0 \leq t_\mathrm{f}.
\end{aligned}
\tag{8.1}
$$

The parameterized initial and terminal constraints of section 5.2.1 are used to make sure that the transition trajectory starts and ends on the corresponding pumping cycle. The parameters $\xi_0 \in [0, 1]$ and $\xi_f \in [0, 1]$ are included as optimization variables, which means that the solver can choose the optimal exit and entry points on the two pumping cycles. As opposed to (7.1), there is no terminal position inequality. Also, instead of the nominal landing cost L, the transition cost K appears in the cost function.

Table 8.1 shows the homotopy schedule for the transition trajectory. The homotopy parameter υ is used in the transition homotopy step to switch from the tracking cost T to the transition cost K. The tether homotopy step is excluded because it was not possible to find a feasible solution for the transition trajectory for any tether drag model except for the trivial one.

Table 8.1.: Transition homotopy schedule.

homotopy step	initial	tracking	transition	final
objective	×	×	$T \xrightarrow{\upsilon} K$	×
dynamics	×	$\mathbf{F}_{\mathrm{FA}} \xrightarrow[\mathbf{F}_{\mathrm{F}}]{\gamma} \mathbf{F}_{\mathrm{A}}$	×	×
constraints	×	×	×	×

An example for a transition trajectory is shown in Fig. 8.1. It shows the transition of the AWES from the initial pumping cycle to another pumping cycle that is approximately half-way to the ground-station. The transition is performed from the reel-in phase of the initial pumping cycle to the reel-in phase of the terminal pumping cycle. The shape of the transition trajectory is dictated by the transition cost K, which penalizes node accelerations and controls. As can be seen most prominently for the red kite in Fig. 8.1, the transition trajectory follows the initial and terminal pumping cycle for a while at the start and the end of the transition trajectory. In between, it describes a smooth inward curve. This curving motion changes the secondary tethers orientation with respect to their respective relative velocities. This is not relevant for the trivial tether drag model that is used to compute the solution shown in Fig. 8.1, but it seems very likely that the optimal trajectory for a more sophisticated tether drag model would be influenced by this effect. This argument is reinforced by the fact that the solver is unable to find a feasible solution for any but the trivial tether drag model. It might be the case that this is just due to poor tuning, but it seems more plausible that the initial guess generation for transition trajectories needs to be changed in order to find feasible solutions for the projected and equivalence tether drag model.

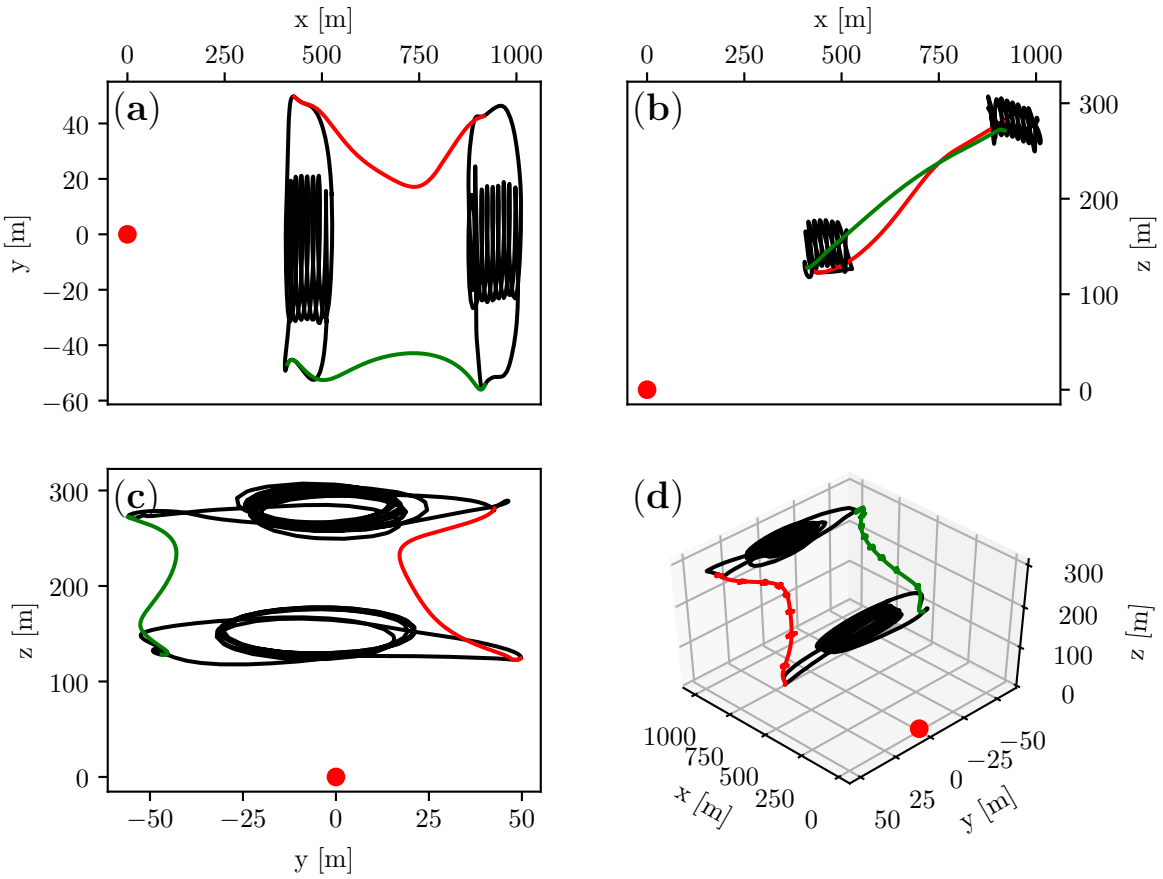

Figure 8.1.: Transition trajectory. Initial and terminal trajectories displayed in black. Position of the ground-station represented by a red dot.

8.2. Emergency Landings with Compromised Flight Behavior

Emergency response strategy (**C**) covers all emergency scenarios that require an immediate landing on account of the compromised AWES's inability to continue on the pumping cycle. As examples for this emergency response strategy, this section will show OCP formulations for three likely scenarios: an impairment of one of the kites' actuators, an on-board battery malfunction an and infliction of structural damage to one of the kites.

8.2.1. Actuator Malfunction

We want to simulate a malfunction of the actuators of kite a. To do so, we use the compromised landing homotopy step to change the bounds of the specific actuator. In

section 5.3, this was formalized as $\mathbf{u}_{\delta,a}^{+} \xrightarrow{\nu} \mathbf{u}_{\delta,a}^{-}$.

As an example, we choose to change the bounds of $\dot{C}_{\mathrm{L},2}$, meaning the lift actuation of the kite at node 2. The bounds after the change are parameterized as

$$\mathbf{u}_{\delta,2}^{-} = \left[f_{\mathrm{c}} \dot{C}_{\mathrm{L,min},2}^{+}, f_{\mathrm{c}} \dot{C}_{\mathrm{L,max},2}^{+}, \dot{\Psi}_{\mathrm{min},2}^{+}, \dot{\Psi}_{\mathrm{max},2}^{+} \right]^{\mathsf{T}}, \tag{8.2}$$

where $f_{\mathrm{c}} \in [0, 1]$ is a parameter that can be used to adjust the degree of the lift actuator malfunction and the superindex + denotes the bounds prior to the change. It should be noted that $f_{\mathrm{c}} = 0$ results in $\dot{C}_{\mathrm{L},2} = 0$ for the whole trajectory, which means that the kite is still able to perfectly stabilize a given $C_{\mathrm{L},2}$.

The resulting OCP formulation is as follows:

$$\begin{aligned}
\underset{\mathbf{x}, E, \boldsymbol{\lambda}, \mathbf{u}, t_{\mathrm{f}}, \mathbf{p}}{\text{minimize}} \quad & \eta T + (1-\eta) L + G \\
\text{subject to} \quad & \mathbf{0} = \mathbf{c}_{\mathrm{col}}(\mathbf{x}_{i,j}, E_{i,j}, \boldsymbol{\lambda}_{i,j}, \mathbf{u}_i, t_{\mathrm{f}}, \mathbf{p}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_{-}, \\
& \mathbf{0} = \mathbf{c}_{\mathrm{con}}(\mathbf{x}_{i+1,0}, E_{i+1,0}, \mathbf{x}_{i,n}, E_{i,n}, \mathbf{p}), && \forall i \in \mathcal{I}^{-}, \\
& \mathbf{0} = \hat{\mathbf{x}}_0(\xi_0) - \mathbf{x}_{0,0}, \\
& 0 = E_{0,0}, \\
& 0 \leq h_{\mathrm{acc}}(\ddot{\mathbf{q}}_{a,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, a \in \mathcal{A}, \\
& 0 \leq h_{\mathrm{ts}}(\lambda_{k,i,j}, l_{k,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_{-}, k \in \mathcal{T}, \\
& 0 \leq h_{\mathrm{acol}}(\mathbf{q}_{a_1,i,j}, \mathbf{q}_{a_2,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, (a_1, a_2) \in \mathcal{A}_{\mathrm{com}}, \\
& 0 \leq h_{\mathrm{tp}}(\mathbf{q}_{p,\bar{m},n}), && \forall p \in \mathcal{N}, \\
& \mathbf{0} \leq \boldsymbol{\lambda}, \\
& \mathbf{u}_{\mathrm{min}}(\mathbf{p}) \leq \mathbf{u} \leq \mathbf{u}_{\mathrm{max}}(\mathbf{p}), \\
& \mathbf{x}_{\mathrm{min}} \leq \mathbf{x} \leq \mathbf{x}_{\mathrm{max}}, \\
& \mathbf{p}_{\mathrm{min}} \leq \mathbf{p} \leq \mathbf{p}_{\mathrm{max}}, \\
& 0 \leq t_{\mathrm{f}}.
\end{aligned} \tag{8.3}$$

In contrast to (7.1) and (8.1), ξ_0 is not an optimization variable but a fixed constant. Other than that, the OCP formulation is identical to (7.1). A further difference to the nominal landing trajectory appears in the homotopy schedule, which can be seen in table 8.2. As already explained, we use the compromised landing homotopy step to change the constraints of the OCP. Due to this constraint change, the bounds on $\mathbf{u}$ depend on $\mathbf{p}$. In order to compute a solution to (8.3), we need to choose values for ξ_0 and f_{c} that best reflect the problem we want to solve. Since it makes little sense to present solutions for arbitrary choices of ξ_0 and f_{c}, we instead vary both parameters within their specified bounds to investigate for which combinations there exist feasible trajectories.

The results of this can be seen in Fig. 8.2. Each step in ξ_0 is represented by a bar. The blue part of the bar represents the values of f_{c} for which there exists a feasible trajectory, while the red part represents the values of f_{c} for which there is no feasible

Table 8.2.: Actuator malfunction homotopy schedule.

homotopy step	initial	tracking	nom. landing	com. landing	tether	final
objective	×	×	$T \xrightarrow{\eta} L$	×	×	×
dynamics	×	$\mathbf{F}_{\mathrm{FA}} \xrightarrow[\mathbf{F}_{\mathrm{F}}]{\gamma} \mathbf{F}_{\mathrm{A}}$	×	×	$\mathbf{F}^{\mathrm{t}}_{\mathrm{TD}} \xrightarrow{\tau} \mathbf{F}^{\mathrm{e}}_{\mathrm{TD}}$	×
constraints	×	×	×	$\mathbf{u}^{+}_{\delta,2} \xrightarrow{\nu} \mathbf{u}^{-}_{\delta,2}$	×	×

trajectory. If a bar is completely red, this means that the solver is not able to find a feasible trajectory for any f_{c}. The dashed black line at $\xi_0 = 0.7$ further divides the parameterized initial trajectory into reel-out and reel-in phase. The two points $\xi_0 = 0.0$ and $\xi_0 = 1.0$ are redundant due to the periodicity of the parameterized initial trajectory. The parameter ξ_0 is varied in steps of 0.025, amounting to 41 steps in total. The step size for f_{c} is 0.5 %.

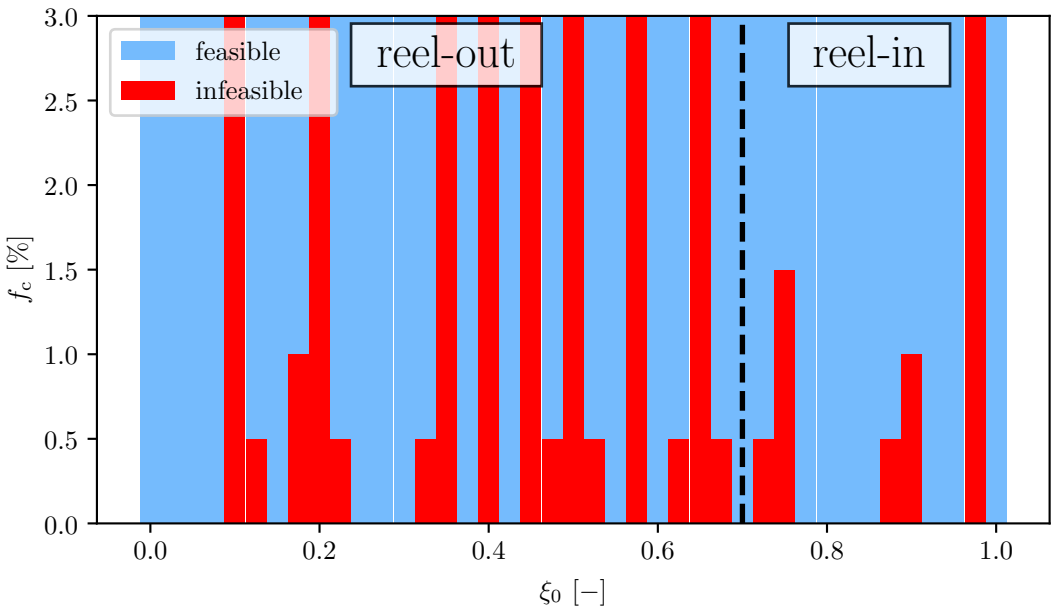

Figure 8.2.: Landing trajectory with compromised actuation. Finding feasible trajectories when varying f_{c} and ξ_0.

Out of the 41 initial conditions, we are able to find feasible solutions for 32. For all of these 32, the smallest possible f_{c} for which we can still find a feasible solution is below 1.5 %. Hence, one can say that if it is possible to find a feasible solution for any f_{c}, then we can usually also find a feasible solution for a very small f_{c}. For 22 out of the

32 trajectories, we can even find a solution for $f_\mathrm{c} = 0$. This leads us to two conclusions: Firstly, the bottle-neck of OCP (8.3) is not the compromised actuator homotopy step, but the choice of the initial condition. This seems reasonable since varying ξ_0 results in a large variety of different initial conditions. Secondly, it makes a huge difference whether $f_\mathrm{c} = 0$ or $f_\mathrm{c} \ll 1.0$. This is also to be expected, since setting $f_\mathrm{c} = 0$ restricts the space of possible trajectories much more than $f_\mathrm{c} \ll 1.0$.

When looking at the separation of reel-in and reel-out phase, we can make another observation: In the reel-out phase, 28.7 % of the bars' surface is marked infeasible, while in the reel-in phase it is only 8.5 %[1]. This is because the initial guesses of section 5.4 continue the AWES's reel-in phase nicely for $\xi_0 \in [0.7, 1.0)$. For the reel-out phase ($\xi_0 \in [0.0, 0.7)$) however, the initial guesses clash with the node velocities of the parameterized initial trajectory. Hence, we can expect better results for the reel-out phase, after finding a way to incorporate the node velocities of $\hat{\mathbf{x}}_0(\xi_0)$ into the initial guess generation.

8.2.2. On-Board Battery Malfunction

We have given an explanation of how the kites' on-board batteries can be modeled in section 4.5. Section 5.2.2 then elaborated on how this model can be implemented into an OCP as the minimum battery energy constraint $E_{\mathrm{B},i} - E_{\mathrm{B,f}} \geq 0 \; \forall i \in \mathcal{I}_-$. This constraint is added to (8.3) to yield the following formulation:

$$
\begin{aligned}
\underset{\mathbf{x}, E, \boldsymbol{\lambda}, \mathbf{u}, t_\mathrm{f}, \mathbf{p}}{\text{minimize}} \quad & \eta T + (1-\eta) L + G \\
\text{subject to} \quad & \mathbf{0} = \mathbf{c}_\mathrm{col}(\mathbf{x}_{i,j}, E_{i,j}, \boldsymbol{\lambda}_{i,j}, \mathbf{u}_i, t_\mathrm{f}, \mathbf{p}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_-, \\
& \mathbf{0} = \mathbf{c}_\mathrm{con}(\mathbf{x}_{i+1,0}, E_{i+1,0}, \mathbf{x}_{i,n}, E_{i,n}, \mathbf{p}), && \forall i \in \mathcal{I}^-, \\
& \mathbf{0} = \hat{\mathbf{x}}_0(\xi_0) - \mathbf{x}_{0,0}, \\
& 0 = E_{0,0}, \\
& 0 \leq E_{\mathrm{B},i}(\mathbf{p}) - E_{\mathrm{B,f}}, && \forall i \in \mathcal{I}_-, \\
& 0 \leq h_\mathrm{acc}(\ddot{\mathbf{q}}_{a,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, a \in \mathcal{A}, \\
& 0 \leq h_\mathrm{ts}(\lambda_{k,i,j}, l_{k,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_-, k \in \mathcal{T}, \\
& 0 \leq h_\mathrm{acol}(\mathbf{q}_{a_1,i,j}, \mathbf{q}_{a_2,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, (a_1, a_2) \in \mathcal{A}_\mathrm{com}, \\
& 0 \leq h_\mathrm{tp}(\mathbf{q}_{p,\bar{m},n}), && \forall p \in \mathcal{N}, \\
& \mathbf{0} \leq \boldsymbol{\lambda}, \\
& \mathbf{u}_\mathrm{min} \leq \mathbf{u} \leq \mathbf{u}_\mathrm{max}, \\
& \mathbf{x}_\mathrm{min} \leq \mathbf{x} \leq \mathbf{x}_\mathrm{max}, \\
& \mathbf{p}_\mathrm{min} \leq \mathbf{p} \leq \mathbf{p}_\mathrm{max}, \\
& 0 \leq t_\mathrm{f}.
\end{aligned}
\tag{8.4}
$$

[1] We use the convention that $\xi_0 = 0.7$ is part of the reel-in phase, $\xi_0 = 0.0$ is part of the reel-out phase and $\xi_0 = 1.0$ is omitted due to its redundancy.

The "activation" of the minimum battery constraint in the compromised landing homotopy step is added to the homotopy schedule shown in table 8.3. The homotopy parameter ν is used to switch from $\frac{\mathrm{d}}{\mathrm{d}t}E_\mathrm{B} = 0$ to $\frac{\mathrm{d}}{\mathrm{d}t}E_\mathrm{B} = \dot{E}_B$ as explained in section 5.3. Using sensible values (cf. appendix A.1) for the on-board battery parameters, one can

Table 8.3.: On-board battery malfunction homotopy schedule.

homotopy step	initial	tracking	nom. landing	com. landing	tether	final
objective	×	×	$T \xrightarrow{\eta} L$	×	×	×
dynamics	×	$\mathbf{F}_\mathrm{FA} \xrightarrow[\mathbf{F}_\mathrm{F}]{\gamma} \mathbf{F}_\mathrm{A}$	×	×	$\mathbf{F}^\mathrm{t}_\mathrm{TD} \xrightarrow{\tau} \mathbf{F}^\mathrm{e}_\mathrm{TD}$	×
constraints	×	×	×	$0 \xrightarrow{\nu} \dot{E}_\mathrm{B}$	×	×

use equation (4.50) to calculate

$$E_{\mathrm{B},0} = N_\mathrm{B}\nu_\mathrm{fr}\nu_\mathrm{c}Q_\mathrm{B}U_\mathrm{B} \tag{8.5}$$
$$= 15 \cdot 1.0 \cdot 0.7 \cdot 18 \text{ kC} \cdot 3.7 \text{ V} \tag{8.6}$$
$$= 699.3 \text{ kJ}, \tag{8.7}$$

as the initial energy of the battery, assuming that it is fully charged. Under the assumption that δ_L, $\dot{\delta}_\mathrm{L}$, δ_Ψ and $\dot{\delta}_\Psi$ are meeting their respective maximum bounds, we can calculate $\dot{E}_{\mathrm{B,act,max}} \approx 48$ W by plugging the values into

$$\dot{E}_\mathrm{B,act} = M_\mathrm{L}\dot{\delta}_\mathrm{L} + M_\phi\dot{\delta}_\Psi. \tag{8.8}$$

Adding the constant power needed for the on-board controller and the rest of the electronics (approximately 60 W) we arrive at $\dot{E}_\mathrm{B,max} \approx 108$ W.

Assuming further a final time for the landing trajectory of 60 s, we get a rough estimate of 6.5 kJ for the upper bound of the energy that is needed for a landing. This amounts to only 0.9 % of the full battery energy of $E_{\mathrm{B},0} = 699.3$ kJ. Even when scaling the battery down from fifteen cells to one, it still only makes up 7.1 % of $E_{\mathrm{B},0}$.

The results of these calculations are supported by the optimization results, where the minimal battery energy constraint does not becomes active at any point of the trajectory for $\nu_\mathrm{fr} \geq 1$ %. Hence, we can conclude that in the event of a malfunction of one of the on-board battery's generators, there is enough time to continue the trajectory until the optimal exit point and employ emergency response strategy (**B**), meaning that the landing trajectory is computed according to the homotopy procedure laid out in chapter 7.

8.2.3. Structural Damage

In case of a bird-strike or a similar event, kite a can sustain structural damage that permanently alters its aerodynamic behavior. This can be modeled by changing the

upper and lower bounds on the lift coefficient $C_{\mathrm{L},a}$ and the roll angle Ψ_a. In section 5.3, this is formalized as

$$\mathbf{x}_{\delta,a}^{+} \overset{\nu}{\to} \mathbf{x}_{\delta,a}^{-}. \tag{8.9}$$

This model completely neglects the dynamic part of the incidence that causes the structural damage. Staying with the example of bird-strike, the model can capture the changed aerodynamic properties of the kite, but not the dynamic perturbation to the system that goes along with it. Hence, the model should only be applied in cases where the perturbation is negligible.

A change in bounds from $\mathbf{x}_{\delta,a}^{+}$ to $\mathbf{x}_{\delta,a}^{-}$ is illustrated in Fig. 8.3, showing the three sets $\mathcal{D} \in \mathcal{W} \in \mathcal{P}$. The set $\mathcal{P} \in \mathbb{R}^2$ is the nominal range of $[C_{\mathrm{L},a}, \Psi_a]$. The set $\mathcal{D} \in \mathbb{R}^2$ is the assumed range for a damaged kite. Lastly, set $\mathcal{W} \in \mathbb{R}^2$ is the real range for the damaged kite. Sticking with the notation of section 5.3, we only consider rectangular sets $\mathcal{D}$ that can be parameterized by a vector

$$\mathbf{x}_{\delta,a}^{-} = \left[f_{\mathrm{c}} C_{\mathrm{L,min},a}^{+}, f_{\mathrm{c}} C_{\mathrm{L,max},a}^{+}, f_{\mathrm{c}} \Psi_{\mathrm{min},a}^{+}, f_{\mathrm{c}} \Psi_{\mathrm{max},a}^{+} \right]^{\mathsf{T}}, \tag{8.10}$$

where $f_{\mathrm{c}} \in [0,1]$ is a parameter used to adjust the degree of the structural damage. Additionally, the parameterized initial condition has to be reformulated as $\hat{\mathbf{x}}_0^{-} - \mathbf{x}_{0,0}^{-}$ to exclude $C_{\mathrm{L},a}$ and Ψ_a from the constraint (cf. (5.64)). Otherwise, there is no guarantee that $[\hat{C}_{\mathrm{L},0,a}, \hat{\Psi}_{0,a}] \in \mathcal{D}$ for all $\xi_0 \in [0,1]$, which is necessary to find a feasible solution.

The resulting OCP then reads as follows:

$$\begin{aligned}
\underset{\mathbf{x}, E, \boldsymbol{\lambda}, \mathbf{u}, t_{\mathrm{f}}, \mathbf{p}}{\text{minimize}} \quad & \eta T + (1-\eta) L + G \\
\text{subject to} \quad & \mathbf{0} = \mathbf{c}_{\mathrm{col}}(\mathbf{x}_{i,j}, E_{i,j}, \boldsymbol{\lambda}_{i,j}, \mathbf{u}_i, t_{\mathrm{f}}, \mathbf{p}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_{-}, \\
& \mathbf{0} = \mathbf{c}_{\mathrm{con}}(\mathbf{x}_{i+1,0}, E_{i+1,0}, \mathbf{x}_{i,n}, E_{i,n}, \mathbf{p}), && \forall i \in \mathcal{I}^{-}, \\
& \mathbf{0} = \hat{\mathbf{x}}_0^{-}(\xi_0) - \mathbf{x}_{0,0}^{-}, \\
& 0 = E_{0,0}, \\
& 0 \le h_{\mathrm{acc}}(\ddot{\mathbf{q}}_{a,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, a \in \mathcal{A}, \\
& 0 \le h_{\mathrm{ts}}(\lambda_{k,i,j}, l_{k,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}_{-}, k \in \mathcal{T}, \\
& 0 \le h_{\mathrm{acol}}(\mathbf{q}_{a_1,i,j}, \mathbf{q}_{a_2,i,j}), && \forall i \in \mathcal{I}, j \in \mathcal{J}, (a_1, a_2) \in \mathcal{A}_{\mathrm{com}}, \\
& 0 \le h_{\mathrm{tp}}(\mathbf{q}_{p,\bar{m},n}), && \forall p \in \mathcal{N}, \\
& \mathbf{0} \le \boldsymbol{\lambda}, \\
& \mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}, \\
& \mathbf{x}_{\min}(\mathbf{p}) \le \mathbf{x} \le \mathbf{x}_{\max}(\mathbf{p}), \\
& \mathbf{p}_{\min} \le \mathbf{p} \le \mathbf{p}_{\max}, \\
& 0 \le t_{\mathrm{f}}.
\end{aligned} \tag{8.11}$$

The OCP formulation is identical to (8.3), but for the change in the parameterized initial condition and the fact that the bounds on $\mathbf{x}$, and not on $\mathbf{u}$, depend on $\mathbf{p}$. Different is

also the homotopy schedule that is shown in table 8.4, where the compromised landing homotopy step is used to change the bounds from $\mathbf{x}_{\delta,2}^{+}$ to $\mathbf{x}_{\delta,2}^{-}$. The set $\mathcal{D}$ is a design

Table 8.4.: Structural damage homotopy schedule.

homotopy step	initial	tracking	nom. landing	com. landing	tether	final
objective	×	×	$T \xrightarrow{\eta} L$	×	×	×
dynamics	×	$\mathbf{F}_{\mathrm{FA}} \xrightarrow[\mathbf{F}_{\mathrm{F}}]{\gamma} \mathbf{F}_{\mathrm{A}}$	×	×	$\mathbf{F}_{\mathrm{TD}}^{\mathrm{t}} \xrightarrow{\tau} \mathbf{F}_{\mathrm{TD}}^{\mathrm{e}}$	×
constraints	×	×	×	$\mathbf{x}_{\delta,2}^{+} \xrightarrow{\nu} \mathbf{x}_{\delta,2}^{-}$	×	×

parameter of (8.11), but how should it be chosen to yield the best results? Suppose a damaged kite has a given real constraint set $\mathcal{W}$ for $[C_{\mathrm{L},a}, \Psi_a]$. For this specific $\mathcal{W}$, $\mathcal{D}$ has to be chosen such that $\mathcal{D} \subseteq \mathcal{W}$ in order to ensure that the trajectory computed for $\mathcal{D}$ is also feasible for $\mathcal{W}$. We define $\mathcal{B}$ as the set of all possible constraint sets for $[C_{\mathrm{L},a}, \Psi_a]$ that can arise from structural damage to the kite. The ideal set $\mathcal{D}^*$ must fulfill the condition

$$\mathcal{D}^* \subseteq \mathcal{W} \; \forall \; \mathcal{W} \in \mathcal{B}. \tag{8.12}$$

If condition (8.12) holds, then the solution of (8.11) would be a feasible trajectory for all possible structural damages with their corresponding $\mathcal{W} \in \mathcal{B}$. Set $\mathcal{W}$ can become almost arbitrarily small (in the sense of the set's Jordan measure) depending on the type and severity of the structural damage the kite has sustained. A $\mathcal{D}^*$ that fulfills condition (8.12) will most likely make (8.11) unfeasible. Hence, using $\mathcal{D}^*$ does not seem reasonable, even if there was a way to compute it. Thus, we have to find a trade-off: make $\mathcal{D}$ as small as possible to account for condition (8.12), but large enough for there to exist a feasible solution to (8.11).

With the given parameterization of equation (8.10), we want to find out how the optimal solution behaves when decreasing f_{c} and therefore also the size of $\mathcal{D}$. To do so, ξ_0 is set to 0.7, which is chosen arbitrarily from the values for ξ_0 for which the solver is able to find feasible solutions in the compromised actuation scenario (cf. section 8.2.1). Taking steps of 0.05 in f_{c}, it is possible to find feasible solutions for a range between 1.0 and 0.3.

Fig. 8.4 shows a comparison of the maximum main tether tension, the optimization time, the average terminal node velocity and the maximum invariant for this range of f_{c}. Since the feasible set of (8.11) shrinks when f_{c} is decreased, we expect the terminal node velocity to increase with a decreasing f_{c}. As Fig. 8.4 shows, this is indeed the case. With a decreasing f_{c}, the movement of the kite at node 2 gets more and more restricted, meaning that it is not able to perform the necessary breaking maneuvers anymore. Nevertheless, the average terminal node velocity does not exceed 1.7 $^{\mathrm{m}}/_{\mathrm{s}}$. We

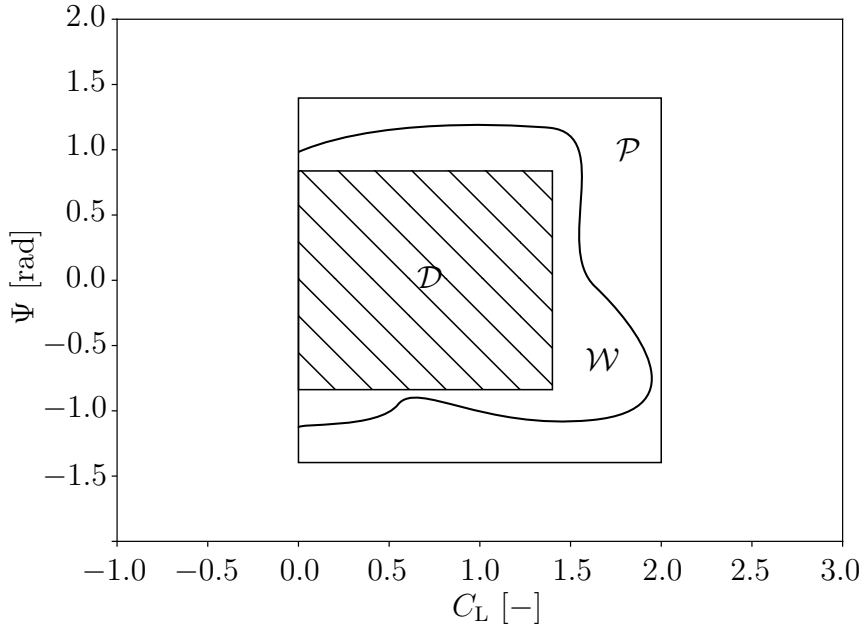

Figure 8.3.: Landing trajectory with structural damage. Illustration of structural damage modeling with assumed constraint set $\mathcal{D}$, real constraint set $\mathcal{W}$ and nominal constraint set $\mathcal{P}$.

can observe that the terminal node velocity first increases linearly until a value of $f_c = 0.6$ with a slope of about 0.33 $^{m}/_{s}$. From $f_c = 0.5$ to 0.3, the terminal node velocity then increases exponentially. The same pattern can be observed when looking at the maximum invariant. Between $f_c = 1.0$ and 0.75, the invariant is constant, followed by a decrease from 0.8 to 0.55. From 0.6 to 0.3 however, there is a strong increase in the invariant, rising to 230 % of the value for $f_c = 1.0$. The terminal node velocities together with the maximum invariants both show the same pattern: At first, the decrease in f_c has little effect on the landing trajectory, but then the solver starts to struggle increasingly with the task of finding a solution that has small invariants and a low terminal node velocity. The maximum main tether tension is one order of magnitude greater than the values for the nominal landing trajectories (cf. Fig 7.8). This is also the case for $f_c = 1$, which means that it must be due to the non-optimal choice of ξ_0. As f_c decreases, the AWES is not able to perform its breaking maneuver as well as before, leading to increasing terminal velocities. Since the maximum tether tension is associated with the breaking maneuver, the less aggressive breaking maneuvers also result in a decrease of the maximum main tether tension. For the optimization time, we expect a similar pattern as for the terminal node velocities or the invariants. This is however not the case. While the optimization time varies and has indeed its maximum at $f_c = 0.3$, there is no inverse correlation between f_c and the optimization time.

This chapter served as an overview of a number of different emergency trajectories, including a transition trajectory and several different emergency landings with compromised

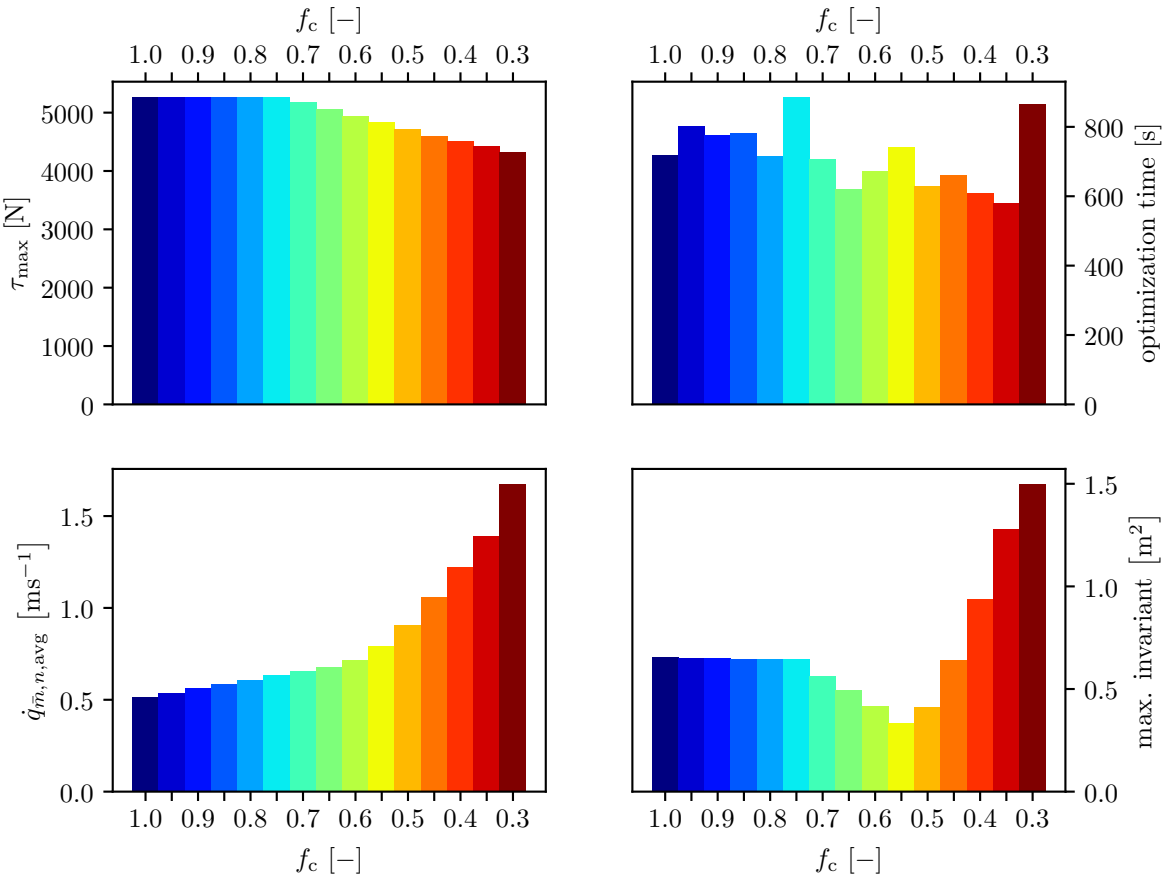

Figure 8.4.: Landing trajectory with structural damage. Sensitivity to change in f_c. Comparison of maximum tether tension, optimization time, average terminal node velocity and maximum invariants.

flight behavior. We have shown that the introduction of the projected or equivalence tether drag model is the bottle-neck of the homotopy scheme when solving for transition trajectories. Whether the solver is able to find a feasible trajectory for an emergency landing with compromised actuation depends on the choice of ξ_0. When starting during the reel-in phase of the parameterized initial trajectory, the solver is able to find feasible trajectories more often than when starting during the reel-out phase. We discovered that a broken on-board generator on one of the kites is no reason for an immediate emergency landing, but that there is enough time to employ emergency landing strategy (**B**). We then studied how the optimal solution for a given ξ_0 changes when shrinking the OCP's feasible set, thereby simulating structural damage to the kites. When shrinking the bounds on one of the kites lift coefficient, the effect on the optimal solution is first small, but then increases more and more until the solver is not able anymore to find

a feasible solution. Section 8.2 has shown that it is indeed possible for the kites of a dual-kite AWES to indirectly control one another in case of an emergency that renders one of the kites actuation compromised.

9. Conclusions and Outlook

In this thesis we set out to construct a general and modular framework that can be used to find and analyze a variety of multi-kite emergency trajectories. In order to do so, we first introduced a set of hierarchical emergency response concepts, each tailored to deal with different types of emergency scenarios. Then we formulated a comprehensive list of modular building blocks that can be used to construct cost functions, constraints and homotopy schemes for multi-kite emergency OCPs. Afterwards, we have shown how this framework can be used to model a variety of different emergency scenarios and to compute the corresponding optimal landing and transition trajectories. The current chapter shall serve to state some final conclusions about the results of this thesis and then give an outlook regarding future research.

Conclusions

The contribution of this thesis is split into two parts: The first part is the formulation of a number of OCPs and the analysis of their solutions. For nominal landing trajectories, we have identified core issues regarding how the optimal solution depends on the tether model and the solution's sensitivity to a change in the reference wind speed. For landing trajectories with compromised flight behavior, we have gained insights into how the initial point of the trajectory determines whether the solver can find a feasible trajectory or not, and how the quality of the landing trajectory decreases when shrinking the feasible set of the OCPs, thereby simulating damaged kites. We also found that a broken kite can indeed be indirectly controlled by the rest of the AWES, thereby making it possible to perform emergency landings with compromised actuation or structural damage on one of the kites. This thesis presents for the first time optimal transition and landing trajectories for dual-kite AWESs.

The second part is the construction of a modular framework that can be used to formulate and solve OCPs related to multi-kite emergency trajectories. The strength of this framework is its flexibility with regard to different models, system configurations and problem formulations. While this thesis focuses on dual-kite configurations with ground-based power generation, the framework can be adapted with moderate effort in the following ways:

- The model can be changed from the 3 DOF point-mass model to a more elaborate 6 DOF model. It is also possible to introduce new tether drag formulations as well as more elaborate tether and actuator models.

- The system configuration can be changed from a dual-kite setup to triple kites or more complex configurations. The framework is also downwards compatible for single kites.

- The homotopy schedule can be adapted by adding or removing any homotopy step that can be formalized in the notation of section 2.4.

- The framework can be modified to comply with on-board power generation instead of ground-based power generation.

- New constraints and cost function components can be added.

The combination of these five points makes it possible to adapt the framework to a large variety of different problem formulations that must not even be connected to emergency landings. A good example is the computation of starting trajectories, which can be implemented within the same framework by adding appropriate constraints, cost terms and homotopy steps. A slight restraint to this flexibility is the necessity to derive new initial guesses for each problem formulation. However this procedure is simplified by the initial and tracking homotopy steps, which generate a feasible initial guess out of a consistent one.

Outlook

In order to improve the emergency landing trajectories that have been presented in this thesis, the first future adaptation should be the change to a 6 DOF model, which was omitted from this thesis due to time constraints.

It is further necessary to improve the automatic initial guess generation in such a way that the parameterized initial and terminal constraints hold for all states of the initial guess, not just the main tether length and node positions. Most notably, it should hold for the node velocities in order to be able to find feasible solutions for all values of ξ_0. This can be done by decoupling the AWES's nodes' motion into a translational and a rotational component with respect to the main tether direction and then parameterizing the node positions and velocities in a way that ensures that the initial guesses have a vanishing invariant. When we interpolate the node positions and velocities as polynomials, we must make sure that they are of sufficiently high degrees such that we can impose all necessary boundary conditions. Within section 5.4, we have started this approach by interpolating the translational motion as a third degree polynomial and the rotational motion as a linear function. This performs well for nominal landing trajectories, but can produce problems for certain values of ξ_0 for compromised landings. Thus, we propose to interpolate both the translational and rotational motion of the nodes as fifth degree polynomials, making it possible to enforce boundary conditions on position, velocity and acceleration of the nodes.

Another open task is finding a suitable OCP formulation to solve for the second part of the landing trajectories. Since we cannot assume anymore that the tethers are permanently tight during the second part of the landing procedure, we need a new tether model that can simulate tether sag. It also needs to be addressed what exactly happens to the secondary tethers during the second part of the landing, as they cannot be reeled in by the generator like the main tether.

On a more theoretical note, further research needs to be done regarding the homotopy strategy presented in section 2.4. Although the strategy works well in practice, we need to find a way to provide convergence guarantees. One approach is to try to formulate conditions on the choice of the homotopy's starting function that guarantee that the solver follows the homotopy path. A promising lead of how to do this is shown in [40], which presents conditions for globally convergent homotopy algorithms for solving nonlinear systems of equations. Since `IPOPT` solves a nonlinear root-finding problem within each iteration, it should be possible to connect these results to the homotopy strategy of section 2.4.

It is useful to incorporate some notion of robustness into the OCP formulations. In this thesis, we have studied the optimal solution's sensitivity to certain parameter changes and found out that e.g. the optimal nominal landing trajectory is robust to a change in the reference wind speed. Section 7.2.2 discusses how the optimal solution changes with the reference wind speed. However, it would also be interesting to see what happens when taking optimal controls that were computed for a wind speed of 5 $^{\mathrm{m}}/_{\mathrm{s}}$ and use them to simulate a landing with different wind speeds. In that way, we can assess the open-loop robustness of a trajectory with respect to a change in the reference wind speed.

This approach however only relies on empirical evidence. We need to find a way to derive actual robustness guarantees for the given OCP formulations or better yet, alter the OCP formulations in such a way that we directly solve for the most robust feasible trajectory. Methods of how to do this can be found e.g. in [15, 35, 34, 36, 20]. Especially for emergency landing trajectories with compromised flight behavior, it makes sense to optimize with respect to robustness to modeling errors. While system identification can be used to derive very accurate models for the nominal flight behavior of an AWES, this is not possible for most emergency scenarios. Implementing robust optimal control into the framework of this thesis will most likely take more time than the previously discussed modifications.

It is still an open question how the results of this thesis can be incorporated into a real-time framework for the detection of emergency scenarios. The implementation of the OCPs inside `AWEbox` are meant for offline computations and are therefore not yet optimized for computation time. So far, it seems however very unlikely that emergency trajectories can be computed on-the-fly for a given emergency scenario. Especially considering that we want to expand the OCP formulations to more complex models, the OCPs are too large to solve them fast enough for an on-the-fly computation. A more realistic approach is precomputing the nominal landing trajectory together with a number of emergency trajectories when also the optimal pumping cycle is computed. The trajec-

tories are then stored in memory and loaded when necessary. In this regard, it appears even more necessary to find a way to increase the robustness of the emergency landing trajectories, since we must account for an error between the initial conditions used to precompute the solutions and the actual initial conditions at the time of the emergency.

In section 8.2 we have shown that the two kites of a dual-kite AWES are able to mitigate the compromised actuation of one of the kites. It will be interesting to study how this phenomenon extends to configurations with three or more kites. If it is indeed possible for larger number of kites to completely support and land a broken kite, then multi-kite AWESs can be designed to be much safer and more reliable than single kite AWESs.

Due to the modular notion of the framework presented in this thesis, we have laid a solid foundation for the upcoming tasks in the field of optimal control for multi-kite emergency trajectories. Since the framework is implemented into the upcoming open-source toolbox `AWEbox`, it will also be accessible to other researchers in the field. By doing so, we are hoping to jumpstart a series of cooperations with other researchers to advance the subject of AWES optimization in general and the computation of optimal trajectories in particular.

A. Appendix A

Table A.1.: Model parameters.

parameter	value	unit	meaning
g	9.81	$^{\mathrm{m}}/_{\mathrm{s}^2}$	standard gravity
ρ_{ref}	1.225	$^{\mathrm{kg}}/_{\mathrm{m}^2}$	reference air density
R_{air}	287.053	$^{\mathrm{J}}/_{\mathrm{kg\ K}}$	specific gas constant for air
T_{ref}	288.15	K	reference temperature
Γ	6.5	$^{\mathrm{K}}/_{\mathrm{km}}$	average lapse rate
ρ_{T}, ρ_{s}	1000.0	$^{\mathrm{kg}}/_{\mathrm{m}}$	main/secondary tether density [14]
σ_{mat}	3.6	GPa	tether maximum material stress [11]
u_{ref}	5.0	$^{\mathrm{m}}/_{\mathrm{s}}$	wind speed at reference height [3]
z_0	0.1	m	roughness length [3]
z_{ref}	10.0	m	reference height [3]
C_{D}^0	0.0273	[−]	wing drag coefficient [25]
C_{D}^I	0.02	[−]	induced drag coefficient
S	3.0	m^2	wing surface [27]
b_a	5.5	m	kite wing span [27]
l_{s}	50.0	m	secondary tether length
d_{T}, d_{s}	0.005	m	main/secondary tether thickness
m_p	36.8	kg	kite weight [27]
m_{G}	10.0	t	generator weight
C_{TD}	1.0	[−]	tether drag coefficient [39]
n_{T}	10	[−]	number of tether segments for eq. forces model
N_{B}	15	[−]	number of battery cells
ν_{fr}	1.0	[−]	fraction of full battery charge
ν_{c}	0.7	[−]	conversion efficiency
Q_{B}	18	kC	battery charge
V_{B}	3.7	V	battery voltage
l_{L}, l_{Ψ}	0.2	m	lift/roll actuator length
w_{L}, w_{Ψ}	0.1	m	lift/roll actuator width
$\delta_{\Psi,\mathrm{min}}$, $\delta_{\mathrm{L,min}}$	−20.0	deg	min. roll/lift actuator deflection
$\delta_{\Psi,\mathrm{max}}$, $\delta_{\mathrm{L,max}}$	20.0	deg	max. roll/lift actuator deflection

Table A.2.: Model bounds.

bound	value	unit	meaning
$C_{\mathrm{L,min}}$	0.0	$[-]$	minimum lift coefficient
$C_{\mathrm{L,max}}$	2.0	$[-]$	maximum lift coefficient
Ψ_{min}	-80.0	deg	minimum roll angle
Ψ_{max}	80.0	deg	maximum roll angle
$\dot{C}_{\mathrm{L,min}}$	-5.0	$1/\mathrm{s}$	minimum lift actuation
$\dot{C}_{\mathrm{L,max}}$	5.0	$1/\mathrm{s}$	maximum lift actuation
$\dot{\Psi}_{\mathrm{min}}$	-5.0	deg/s	minimum roll actuation
$\dot{\Psi}_{\mathrm{max}}$	5.0	deg/s	maximum roll actuation
$\ddot{l}_{\mathrm{T,min}}$	-10.0	m/s^2	minimum main tether acceleration
$\ddot{l}_{\mathrm{T,max}}$	10.0	m/s^2	maximum main tether acceleration
z_{min}	10	m	minimum node height
$l_{\mathrm{T,min}}$	0	m	minimum main tether length

Table A.3.: Numerical parameters.

parameter	value	unit	meaning
m	50	[−]	number of collocation intervals
n	4	[−]	degree of the Lagrange polynomial used for collocation
τ_a	1.0	[−]	starting value for τ_0
τ_b	$1 \cdot 10^{-5}$	[−]	intermediate value for τ_0 and τ_f
τ_c	$1 \cdot 10^{-8}$	[−]	final value for τ_f
r_a	$1 \cdot 10^{-4}$	[−]	initial value for r_tol
r_b	$1 \cdot 10^{-8}$	[−]	final value for r_tol
κ	10.0	[−]	Baumgarte stabilization
$\hat{R}_u$	$1 \cdot 10^{-4}$	[−]	control regularization tuning
$\hat{R}_t$	$1 \cdot 10^{-2}$	[−]	time regularization tuning
$\hat{R}_{\ddot{q}}$	0.1	[−]	node acceleration regularization tuning
$\hat{H}_\gamma$	$1 \cdot 10^{3}$	[−]	γ tuning cost
$\hat{H}_\eta$	$1 \cdot 10^{3}$	[−]	η tuning cost
$\hat{H}_\nu$	$1 \cdot 10^{3}$	[−]	ν tuning cost
$\hat{H}_\tau$	$1 \cdot 10^{3}$	[−]	τ tuning cost
$\hat{H}_\upsilon$	$1 \cdot 10^{3}$	[−]	υ tuning cost
$\hat{H}_{\boldsymbol{\theta}}$	$1 \cdot 10^{3}$	[−]	$\boldsymbol{\theta}$ tuning cost
$\hat{\mathbf{T}}$	$\mathrm{diag}(\mathbf{1}) \cdot 10^{-2}$	[−]	tracking homotopy tuning
$\hat{\mathbf{L}}$	$\mathrm{diag}(\mathbf{1}) \cdot 10^{-2}$	[−]	nominal landing homotopy tuning
$\hat{K}_{\ddot{q}}$	1.0	[−]	transition node acceleration regularization tuning
$\hat{K}_u$	0.1	[−]	transition control regularization tuning
c_s	10.0	[−]	tether stress safety factor
A_max	$8 \cdot 10^{-3}$	m^2	tether stress scaling
c_acc	12.0	[−]	max. acceleration factor
c_min	5.0	[−]	anti-collision safety factor
d_s	0.1	m	“stagger” distance
$d_\mathrm{min,1}$	40.0	m	terminal position distance for node 1
$d_\mathrm{min,2}$	80.0	m	terminal position distance for node 2
$d_\mathrm{min,3}$	80.0	m	terminal position distance for node 3
v_T	22.0	m/s	landing velocity guess

B. Appendix B

Code Example B.1: Code example in `python` to produce Fig. 7.4 using `AWEbox`.

```python
# import modules
from AWEbox import *
import logging
# configure level of logging/console output
logging.basicConfig( \
   filemode='w', \
   format='%(message)s', \
   level=logging.DEBUG)

# ================
# SET-UP AND SOLVE
# ================

# make default options object and configure it
options = Options(True) # True refers to internal access switch
options['user_options']['system_model'] \
   ['architecture'] = (1, 2) # dual kites
options['user_options']['trajectory'] \
   ['fixed_params'] = \
   {'l_s':[50.,50.],'diam_t':[5e-3,5e-3],'diam_s':[5e-3,5e-3]} # tether
    parameters
options['user_options']['trajectory'] \
   ['type'] = 'nominal_landing' # trajectory type
options['user_options']['trajectory'] \
   ['transition']['initial_trajectory'] = \
   'initial_dual.p' # param. pumping cycle
options['user_options']['system_model']['kite_dof'] = 3 # 3 DOF
options['nlp']['n_k'] = 50 # number of coll. intervals
options['nlp']['collocation']['d'] = 4 # deg. of lagrange polynomial

# define sweep options
sweep_opts = [(['user_options','tether_drag_model'], \
   ['trivial','simple','equivalence'])] # sweep over tether drag models

# make sweep, run and save
sweep = Sweep( \
   name = 'sweep_example', \
   options = options, \
   sweep_opts = sweep_opts)
```

```
39 sweep.run(final_homotopy_step='tether') # solve until (and including)
       tether homotopy step
40 sweep.save()
41
42 # ==============
43 # POSTPROCESSING
44 # ==============
45
46 # make visualization object and options
47 visualization = Visualization(sweep)
48 visualization_options = Visualization_options()
49
50 # set flags for the desired plots
51 visualization_options['user_options']['flags']
52   ['comparison']['quad'] = True # trajectory plot
53
54 # produce and show plots
55 visualization.plot(visualization_options)
```

Bibliography

[1] A. Adler, A. Bar-Gill, and N. Shimkin. Optimal flight paths for engine-out emergency landing. In *Proceedings of the 24th Chinese Control and Decision Conference*, pages 2908–2915, 2012.

[2] J. Andersson, J. Åkesson, and M. Diehl. CasADi: A symbolic package for automatic differentiation and optimal control. In S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, editors, *Recent Advances in Algorithmic Differentiation*, volume 87 of *Lecture Notes in Computational Science and Engineering*, chapter 27, pages 297–307. Springer, 2012.

[3] C. Archer. An introduction to meteorology for airborne wind energy. In U. Ahrens, M. Diehl, and R. Schmehl, editors, *Airborne Wind Energy*, chapter 5, pages 81–94. Springer, 2013.

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

[5] E. Bakolas, Y. Zhao, and P. Tsiotras. Initial guess generation for aircraft landing trajectory optimization. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pages 2908–2915, Portland (OR), USA, 2011.

[6] G. Bryan. *Stability in Aviation.* McMillan, 1911.

[7] C. Büskens and D. Wassel. The ESA NLP solver WORHP. In G. Fasano and J. Pintér, editors, *Modeling and Optimization in Space Engineering*, volume 73, pages 85–110. Springer New York, 2013.

[8] A. Cherubini, A. Papini, R. Vertechy, and M. Fontana. Airborne wind energy systems. A review of the technologies. *Renewable and Sustainable Energy Reviews*, 51:1461–1476, 2015.

[9] M. Diehl. Airborne wind energy: Basic concepts and physical foundation. In U. Ahrens, M. Diehl, and R. Schmehl, editors, *Airborne Wind Energy*, chapter 1, pages 3–22. Springer, 2013.

[10] M. Diehl and B. Houska. Windenergienutzung mit schnell fliegenden Flugdrachen: Eine Herausforderung für die Optimierung und Regelung. *Automatisierungstechnik*, 57(10):525–533, 2009.

[11] Dyneema Fact Sheets. Ultra high molecular weight polyethylene fiber from DSM dyneema. `https://issuu.com/eurofibers/docs/name8f0d44`, 2010. Accessed: 2018-4-22.

[12] P. Eng. *Path Planning, Guidance and Control for a UAV Forced Landing*. PhD thesis, Queensland University of Technology, 2011.

[13] P. Eng, L. Mejias, R. Walker, and D. Fitzgerald. Guided chaos: Path planning and control for a UAV-forced landing. *IEEE Robotics and Automation Magazine*, 17(2):90–98, 2010.

[14] U. Fechner. *A Methodology for the Design of Kite-Power Control Systems*. PhD thesis, Technische Universiteit Delft, 2016.

[15] J. Gillis. *Practical methods for approximate robust periodic optimal control of nonlinear mechanical systems*. PhD thesis, KU Leuven, 2015.

[16] W. Grimm. Bahnoptimierung für Luft- und Raumfahrzeuge. Skriptum zur Vorlesung, 2004. Accessed: Winter Term 2016/17.

[17] S. Gros, M. Zanon, and M. Diehl. Modeling of airborne wind energy systems in natural coordinates. In U. Ahrens, M. Diehl, and R. Schmehl, editors, *Airborne Wind Energy*, chapter 10, pages 205–218. Springer, 2013.

[18] S. Gros, M. Zanon, and M. Diehl. A relaxation strategy for the optimization of airborne wind energy systems. In *Proceedings of the 12th European Control Conference (ECC)*, pages 1011–1016, Zürich, Switzerland, July 2013.

[19] G. Horn, S. Gros, and M. Diehl. Numerical trajectory optimization for airborne wind energy systems described by high fidelity aircraft models. In U. Ahrens, M. Diehl, and R. Schmehl, editors, *Airborne Wind Energy*, chapter 11, pages 205–218. Springer, 2013.

[20] B. Houska. *Robust Optimization of Dynamic Systems*. PhD thesis, KU Leuven, 2011.

[21] B. Houska and M. Diehl. Optimal control for power generating kites. In *Proceedings of the 9th European Control Conference (ECC)*, pages 3560–3567, 2007.

[22] HSL. A collection of Fortran codes for large scale scientific computation. `http://www.hsl.rl.ac.uk/`. Accessed: 2017-12-20.

[23] J. Koenemann, P. Williams, S. Sieberling, and M. Diehl. Modeling of an airborne wind energy system with a flexible tether model for the optimization of landing trajectories. In *Proceedings of the IFAC World Congress*, pages 11944–11950, 2017.

[24] R. Leuthold, J. De Schutter, E. Malz, G. Litrica, S. Gros, and M. Diehl. Operational regions of a multi-kite AWE system. In *Proceedings of the 16th European Control Conference (ECC)*, 2018. to be published.

[25] G. Licitra. Numerical optimal trajectory for system in pumping mode described by differential algebraic equation (focus on ap2). Internal Document, Ampyx Power B.V., Lulofsstraat 55, 2521 AL The Hague, The Netherlands, 2014.

[26] G. Licitra, S. Sieberling, S. Engelen, P. Williams, R. Ruiterkamp, and M. Diehl. Optimal control for minimizing power consumption during holding patterns for airborne wind energy pumping system. In *Proceedings of the 15th European Control Conference (ECC)*, 2016.

[27] G. Licitra, P. Williams, J. Gillis, S. Ghandchi, S. Sieberling, R. Ruiterkamp, and M. Diehl. Aerodynamic parameter identification for an airborne wind energy pumping system. *IFAC-PapersOnLine*, 50(1):11951–11958, 2017.

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

[29] L. Mejias, D. Fitzgerald, P. Eng, and L. Xi. Forced landing technologies for unmanned aerial vehicles: Towards safer opterations. In L. T. Mung, editor, *Aerial Vehicles*, chapter 21, pages 413–440. InTech, 2009.

[30] S. Meng, J. Xiang, Z. Luo, Y. Ren, and N. Zhuang. A novel trajectory planning strategy for aircraft emergency landing using Gauss pseudospectral method. *Control Theory and Technology*, 12(4):393–401, 2014.

[31] B. Pamadi. *Performance, Stability Dynamics, and Control of Airplanes*. American Institute of Aeronautics and Astronautics, 2nd edition, 2004.

[32] J. Rawlings, D. Mayne, and M. Diehl. *Model Predictive Control. Theory, Computation and Design*, chapter Numerical Optimal Control, pages 491–595. Nob Hill Publishing, 2nd edition, 2017.

[33] S. Russell and P. Norvig. *Artificial Intelligence. A Modern Approach*. Pearson Education Inc., 2010.

[34] J. Sternberg, J. Goit, S. Gros, J. Meyers, and M. Diehl. Robust and stable periodic flight of power generating kite systems in a turbulent wind flow field. In *Proceedings of the 15th IFAC Workshop on Control Applications of Optimization*, 2012.

[35] J. Sternberg, S. Gros, B. Houska, and M. Diehl. Approximate robust optimal control of periodic systems with invariants and high-index differential algebraic systems. In *Proceedings of the 7th IFAC Symposium on Robust Control Design*, pages 690–695, Aalborg, Denmark, June 2012.

[36] J. Sternberg, B. Houska, and M. Diehl. A structure exploiting algorithm for approximate robust optimal control with application to power generating kites. In *Proceedings of the American Control Conference (ACC)*, 2012.

[37] R. Szabolscsi. A new emergency landing concept for unmanned aerial vehicles. *Review of the Air Force Acedemy*, 2(32):5–12, 2016.

[38] P. Tang, S. Zhang, and J. Li. Final approach and landing trajectory generation for civil airplane in total loss of thrust. In *Proceedings of the 3rd International Symposium on Aircraft Airworthiness*, pages 522–528, 2013.

[39] R. von Mises. *Theory of Flight*. Dover Publications, 1959.

[40] L. Watson. Probability-one homotopies in computational science. *Journal of Computational and Applied Mathematics*, 140:785–807, 2002.

[41] A. Wächter and L. Biegler. On the implementation of an interior-point filter line-search alrgorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[42] H. Wu and F. Mora-Camino. Knowledge-based trajectory control for engine-out aircraft. In *Proceedings of the 32nd Digital Avionics Systems Conference*, pages 522–528, East Syracuse (NY), USA, 2013.

[43] K. Wu, Z. Cai, Q. Lin, and Y. Wang. Design of landing trajectory for unpowered aircraft using an *hp*-adaptive Radau pseudospectral method. In *Proceedings of the IEEE Chinese Guidance, Navigation and Control Conference*, pages 2150–2155, Yantai, China, 2014.

[44] A. Zanelli, R. Quirynen, J. Jerez, and M. Diehl. A homotopy-based nonlinear interior-point method for NMPC. In *Proceedings of the 20th World Congress of the International Federation of Automatic Control*, pages 13731–13735, 2017.

[45] M. Zanon, S. Gros, J. Andersson, and M. Diehl. Airborne wind energy based on dual airfoils. *IEEE Transactions on Control Systems Technology*, 21(4):1215–1222, 2013.