FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Optimization of Energy within the Complete Cycle of Airborne Wind Energy Systems (AWES)

Eduardo Cabral Martins Barbosa

Master's Degree in Electrical and Computer Engineering

Supervisor: Maria do Rosário de Pinho

Second Supervisor: Luís Tiago Paiva

November 3, 2024

© Eduardo Cabral Martins Barbosa, 2024

Resumo

Os Sistemas Aerotransportados de Energia Eólica (SAEE) oferecem o potencial de operar a altitudes significativamente mais elevadas do que as turbinas eólicas convencionais, tornando-os mais eficientes e amigos do ambiente. Nesta dissertação, centrámo-nos na otimização das fases de produção e armazenamento de energia dos SAEE utilizando um modelo de simulação dinâmica. O nosso objetivo foi desenvolver estratégias operacionais que maximizem a viabilidade económica e a sustentabilidade ambiental, melhorando assim o desempenho do sistema e o rendimento energético.

Utilizando ferramentas de simulação avançadas, modelámos a aerodinâmica dos sistemas aéreos, bem como a sua dinâmica. O nosso objetivo foi maximizar a captação de energia durante as fases ativas do voo e minimizar as perdas durante a fase de recuperação, otimizando a trajetória e as estratégias de controlo. Os principais tópicos abordados incluem a conceção de um controlador de trajetória para trajetórias ótimas em termos de energia e estratégias para atenuar o vento, a fim de evitar uma carga excessiva no sistema de cabo.

Modelámos as contribuições aerodinâmicas para o ciclo energético do SAEE, utilizando técnicas 2D e 3D simplificadas. Posteriormente, gerámos uma curva de potência total para quantificar o desempenho em várias condições de funcionamento.

Abstract

Airborne Wind Energy Systems (AWES) offer the potential to operate at significantly higher altitudes than conventional wind turbines, making them more efficient and environmentally friendly. In this dissertation, we focused on optimizing the energy generation and storage phases of AWES using a dynamic simulation model. Our aim was to develop operational strategies that maximize both economic viability and environmental sustainability, thus improving system performance and energy yield.

Using advanced simulation tools, we modeled the aerodynamics of both the airborne and tether systems, as well as their mechanical dynamics. Our goal was to maximize power capture during the active phases of flight and minimize losses during the recovery phase by optimizing the path trajectory and control strategies. Key topics covered include the design of a trajectory controller for optimal trajectories in terms of energy and strategies to mitigate the wind to avoid excessive load on the cable system.

We modeled the aerodynamic contributions to the AWES energy cycle in detail, employing 2D and simplified 3D techniques. Subsequently, we generated a full power curve to quantify performance under various operating conditions.

United Nations Sustainable Development Goals (SDGs)

Introduction

This chapter outlines the alignment of the research and findings presented in this dissertation with three key United Nations Sustainable Development Goals (SDGs). The SDGs serve as a universal call to action to end poverty, protect the planet, and ensure that all people enjoy peace and prosperity by 2030. The integration of sustainable engineering practices is crucial to achieving these goals.

SDG 7: Affordable and Clean Energy

Relevance to Kite Power Systems

Kite Power Systems (KPS) contribute significantly to SDG 7 by providing a renewable energy source that can reduce reliance on fossil fuels. The dynamic modeling and optimization of KPS can lead to more efficient energy generation methods, making clean energy more accessible and affordable.

Impact and Innovation

This research focuses on improving the performance and efficiency of KPS. Innovations in aerodynamic modeling, control algorithms, and simulation techniques contribute to the development of sustainable energy technologies. By optimizing the flight paths and energy extraction processes of kites, this work supports the transition to a more sustainable energy infrastructure. The use of advanced simulation tools, such as AWEBox, enhances the precision and reliability of these optimizations, ensuring that KPS can effectively harness wind energy.

SDG 12: Responsible Consumption and Production

Promoting Sustainable Practices

Kite Power Systems have the potential to contribute to more sustainable consumption and production patterns by providing an environmentally friendly alternative to traditional energy sources. The use of kites for energy generation reduces the dependency on non-renewable resources and minimizes environmental impact. This dissertation demonstrates how KPS can be integrated into existing energy systems, promoting a shift towards greener and more sustainable energy consumption.

Efficient Resource Use

The research presented in this dissertation emphasizes the efficient use of resources through advanced simulation and optimization techniques. By maximizing the energy output from kites and minimizing waste, this work supports responsible consumption and production practices. The development of efficient control strategies for KPS ensures that the systems operate at optimal performance, thus saving resources and reducing operational costs.

SDG 13: Climate Action

Mitigating Climate Change

Kite Power Systems represent a clean energy solution that helps mitigate the effects of climate change. By reducing greenhouse gas emissions and reducing the carbon footprint of energy generation, KPS directly contributes to the objectives of SDG 13. This research highlights the potential of KPS to replace or supplement traditional fossil fuel-based energy sources, thus playing a critical role in global climate change mitigation efforts.

Adapting to Climate Impacts

The research presented in this dissertation also addresses the adaptability of KPS to various environmental conditions. By improving the robustness and efficiency of KPS, this work ensures that these systems can be effectively deployed in various climatic regions, supporting climate resilience and adaptation strategies. The flexibility of KPS to operate under different wind conditions makes them a versatile solution for sustainable energy generation in the face of changing climate patterns.

Conclusion

The research and innovations presented in this dissertation align closely with several Sustainable Development Goals of the United Nations, particularly SDG 7 (Affordable and Clean Energy), SDG 12 (Responsible Consumption and Production), and SDG 13 (Climate Action). By advancing the technology and application of Kite Power Systems, this work could contribute to the global effort to create a more sustainable and equitable future.

Through the development of efficient aerodynamic models, robust control algorithms, and advanced simulation tools, this dissertation demonstrates the feasibility and effectiveness of KPS as a sustainable energy solution. The integration of renewable energy solutions, responsible consumption and production practices, and climate action underscore the critical role of engineering in achieving the SDGs.

Acknowledgments

I extend my deepest gratitude to Professor Maria do Rosário de Pinho and Professor Luís Tiago Paiva for their invaluable guidance, support, and encouragement throughout this research. Their expertise and mentorship have been crucial in shaping my academic journey.

I also thank my family, especially my parents, for their unwavering support, and my girlfriend for her love and understanding. Their encouragement has been a constant source of strength.

Eduardo Cabral Martins Barbosa

"Na cama que fizeres,
Na cama que te deitas."

Júlio Pereira

Contents

Acknowledgments **ix**

Nomenclature **xx**

Preface **xxii**

1 Introduction **1**
1.1 Context . . . 1
1.2 Motivation . . . 3
1.3 Objectives . . . 4
1.4 Document Structure . . . 5

2 State of the Art in AWES and Kite Technologies **7**
2.1 Overview of AWES Configurations . . . 8
2.1.1 Ground-Gen AWES . . . 8
2.1.2 Fly-Gen AWES . . . 9
2.1.3 Hybrid AWES . . . 10
2.2 Challenges and Technological Developments . . . 10
2.3 Current Research and Future Directions . . . 11
2.4 Case Study: UPWIND Project . . . 11
2.4.1 Future Prospects . . . 12
2.5 Conclusion . . . 13

3 Dynamic Model of the Kite Power System **15**
3.1 Introduction . . . 15
3.2 Coordinate Systems . . . 15
3.2.1 Global Coordinate System (G) . . . 15
3.2.2 Local Coordinate System (L) . . . 16
3.2.3 Body Coordinate System (B) . . . 17
3.2.4 Coordinate Transformations . . . 17
3.3 Acting Forces . . . 18
3.3.1 Aerodynamic Forces . . . 18
3.3.2 Gravitational Force . . . 19
3.3.3 Tether Tension . . . 19
3.3.4 Mathematical Integration of Forces . . . 20
3.4 Dynamic Equations . . . 20

3.5 Control Variables . . . 21
3.5.1 Tether Acceleration . . . 21
3.5.2 Angle of Attack . . . 21
3.5.3 Roll Angle . . . 22
3.6 Simulation and Control . . . 23

4 Simulation, Integration and Optimization of Kite Dynamics 25
4.1 Introduction . . . 25
4.2 Optimal Control . . . 25
4.3 Discretization Methods . . . 26
4.3.1 Computational Tools for Optimization . . . 27
4.4 Numerical Methods and Solvers . . . 28
4.4.1 Discretization and Implementation . . . 28
4.4.2 Optimization Solvers . . . 29
4.5 Simulation, Integration, and Optimization of Kite Dynamics . . . 29
4.5.1 Initial Simulation with MATLAB and Simulink . . . 30
4.5.2 Optimization with CasADi and MATLAB . . . 30
4.5.3 Transition to AWEBox and use of IPOPT with HSL Solvers . . . 30
4.5.4 Environment Setup and Time Investment . . . 31
4.5.5 Simulation and Optimization Scenarios . . . 31
4.5.6 Advanced Control Strategies . . . 33
4.6 Optimal Control and Optimization Objectives . . . 34
4.6.1 Problem Formulation . . . 34
4.6.2 System Constraints . . . 36
4.7 Conclusion . . . 37

5 Simple Closed-Loop Trajectory 39
5.1 Control and Mathematical Problem for the First Simulation . . . 39
5.1.1 Mathematical Problem . . . 39
5.1.2 Control Objective . . . 40
5.2 Simulation Setup . . . 40
5.2.1 Numerical Details . . . 41
5.3 Results Analysis . . . 43
5.4 Optimized Trajectory Simulation Results . . . 46
5.4.1 Overview . . . 46
5.4.2 Simulation Results . . . 46
5.4.3 Discussion . . . 48
5.4.4 Conclusion . . . 48

6 Closed-Loop Trajectory with Ascending and Retracting Phase 51
6.1 Introduction . . . 51
6.2 Control and Mathematical Problem for the Second Simulation . . . 52
6.2.1 Traction Mode (Reel-Out Phase) . . . 52
6.2.2 Transition Mode . . . 53
6.2.3 Retraction Mode (Reel-In Phase) . . . 53
6.2.4 Numerical Details . . . 54
6.3 Implementation and Results . . . 55

6.3.1 Control Variables Over Time 56
6.3.2 Power Output Over a Complete Cycle 56
6.3.3 Trajectory Plot of the Kite 58
6.3.4 Discussion 62
6.3.5 Conclusion 62

7 Conclusions and Future Work **64**
7.1 Conclusions 64
7.2 Future Work 66

A Appendix A **70**

B Appendix B **73**

List of Figures

1.1 UPWIND [1] kite model prototype. 3
1.2 Fundamental idea behind Airborne Wind Energy systems with a ground-based generator [8]. 4

2.1 Schematic of Ground-Gen AWES [11]. 8
2.2 Schematic of Fly-Gen AWES [13]. 9
2.3 Institutions With Airborne Wind Energy Projects [18] 12

3.1 Global and Local Coordinate Systems - Referential Representation. 16
3.2 Global and Local Coordinate Systems - Kite Representation. 17
3.3 Angle of Attack [21]. 22
3.4 Roll angle control mechanism [21]. 22

4.1 Comparison of Ipopt performance over various linear solvers using the two-dimensional partial differential equation test problem set. [26] 28

5.1 Tether Length (r_t): Length of tether deployed, indicating kite altitude. 43
5.2 Tether Reel-out Speed (v_t): Speed of tether extension/retraction. 43
5.3 Total Energy (e): Total system energy, combining kinetic and potential components. . . . 44
5.4 Deflection rates ($\dot{\delta}_{\text{aileron}}, \dot{\delta}_{\text{elevator}}, \dot{\delta}_{\text{rudder}}$) during the first simulation. These rates represent the time derivatives of the control surface deflections for the aileron, elevator, and rudder. The yaw angle (ψ, green), pitch angle (θ, blue), and roll angle (ϕ, orange) are plotted to illustrate how the kite's orientation changes throughout the first simulation as a result of these deflections. The goal of this first simulation was to test the stability of the system and ensure that the kite followed a closed-loop trajectory. 44
5.5 Trajectory in the XY plane. 46
5.6 Trajectory in the XZ plane. 47
5.7 Trajectory in the ZY plane. 47
5.8 Three-dimensional trajectory. 48

6.1 Tether Length (r_t): Length of tether deployed, indicating kite altitude. 55
6.2 Tether Reel-out Speed (v_t): Speed of tether extension/retraction. 56
6.3 Total Energy (e): Total system energy, combining kinetic and potential components. . . . 56
6.4 Deflection rates ($\dot{\delta}_{\text{aileron}}, \dot{\delta}_{\text{elevator}}, \dot{\delta}_{\text{rudder}}$) for the control surfaces of the kite, representing the rate of change of the aileron, elevator, and rudder deflections. The yaw angle (ψ, green), pitch angle (θ, blue), and roll angle (ϕ, orange) are shown in the graph, illustrating how the kite's orientation changes over time as a result of the deflections. 57

6.5 Trajectory in the XY plane. 58
6.6 Trajectory in the XZ plane. 59
6.7 Trajectory in the ZY plane. 59
6.8 Three-dimensional trajectory. 60
6.9 Detailed three-dimensional trajectory. The trajectory starts at the *initial state* (green) and ends at the *final state* (red), following an upward path and then returning to the initial point. 61

List of Tables

4.1 AWEBox Required Dependencies and Hardware Recommendations 33
4.2 Parameters 36

5.1 Numerical Details of the Optimization Problem used in AWEBox for this simulation. . . 42

6.1 Numerical Details of the Optimization Problem used in AWEBox for this simulation. . . 54

List of Acronyms

SAEE	Sistemas Aerotransportados de Energia Eólica
AWES	Airborne Wind Energy Systems
KPS	Kite Power Systems
PKG	Pumping Kite Generator
LCOE	Levelized Cost of Energy
FG	Flying Generation
GG	Ground Generation
PSD	Power Spectral Density
GWA	Global Wind Atlas
RGL, RLG	Rotation matrix from Global to Local and vice versa

Nomenclature

AR	wing aspect ratio
A	wing reference area of kite [m^2]
a_t	tether reel-out acceleration [m/s^2]
r	tether length [m]
c_D	aerodynamic drag coefficient
ρ	air density [kg/m^3]
c_L	aerodynamic lift coefficient
s	wing span [m]
E	energy produced [Ws]
T	tether tension [N]
$\vec{F}_{\text{aer}}$	aerodynamic force [N]
$\vec{v}_a$	apparent wind velocity [m/s]
$\vec{F}_{\text{drag}}$	drag force [N]
$\vec{v}_w$	wind velocity [m/s]
$\vec{F}_{\text{cent}}$	centrifugal force [N]
$\vec{v}_t$	tether reel-out velocity [m/s]
$\vec{F}_{\text{cor}}$	Coriolis force [N]
$\vec{F}_{\text{lift}}$	aerodynamic lift force [N]
$\vec{u}$	control vector
$\vec{F}_{\text{inert}}$	inertial forces [N]
$\vec{x}$	state vector
$\vec{F}_{\text{th}}$	tether force [N]
α	angle of attack [rad]
g	gravitational acceleration [m/s^2]
ϕ	azimuthal angle [rad]
m	mass [kg]
β	elevation angle [rad]
P	power produced [W]
θ	polar angle [rad]
R90	90° anticlockwise rotation matrix
ψ	roll angle [rad]

Preface

This dissertation is a part of the UPWIND Research Project [1], which focuses on exploring Airborne Wind Energy Systems (AWES). The project emphasizes systems utilizing a tethered wing flying in high-speed crosswind motions and covers several critical areas, including:

- Optimization and control of AWES
- Simulation in both 2D and 3D environments
- Control optimization of kite trajectory

Chapter 1

Introduction

This thesis is part of the *Master's Degree in Electrical and Computer Engineering* program, focusing on the optimal control of Airborne Wind Energy Systems (AWES). The research is carried out within the framework of the UPWIND research project [1], which aims to develop innovative solutions for the capture of high-altitude wind energy using dynamic airborne devices.

The study involves determining optimal control strategies for AWES, using dynamical models to simulate various trajectory scenarios. The objective is to maximize energy capture and minimize operational losses. The optimal control problem is addressed using the direct method, where the problem is transcribed into a non-linear programming problem and solved using the IPOPT solver[2] integrated with Coin-HSL [3], supported by CASADi [4]. In addition, the AWEBox [5] library and HSL solvers are used to enhance the modeling and optimization processes. All computational scripts are run in Python, ensuring efficient handling of simulations and data analysis.

This thesis aims to contribute to the field of renewable energy by optimizing the complete energy cycle of AWES, thereby improving the efficiency and economic viability of the system. The findings are expected to support future advances in AWES technologies and their broader application in sustainable energy solutions.

1.1 Context

The reliance on fossil fuels and non-renewable resources has led to severe environmental and societal impacts, including climate change, environmental degradation, and health issues worldwide. Despite the EU's goals to cut greenhouse gas emissions by 20% from 1990 levels and increase renewable energy and efficiency by 20% by 2020, sustainable energy solutions are urgently needed. Looking to 2024 and beyond, the focus has shifted to reducing carbon emissions further, improving energy storage, and better integrating renewable sources into grids [6].

In recent years, renewable energy has significantly increased its share of global electricity. By 2023, wind energy saw major advancements in technology and capacity, surpassing previous records and integrating digital and automation technologies for improved efficiency. Innovations in harnessing higher-altitude wind, where speeds are stronger and more stable, have driven the development of Airborne Wind Energy Systems (AWES), using kites or drones that reach higher altitudes without large structures [7].

The Pumping Kite Generator (PKG) system, set for major development by 2024, features a tethered airfoil connected to a generator via a winch. The PKG generates electricity as the tether extends during flight and uses a fraction of this energy to retract the tether, making it efficient and cost-effective compared to traditional turbines. Key AWES milestones include achieving grid parity, proving commercial viability, and expanding offshore deployments. Advances in materials science are expected to further enhance the performance and durability of AWES, positioning them as future renewable energy cornerstones.

Ongoing research and deployments will drive continuous improvements in AWES design and protocols, reducing the environmental impact of traditional energy sources and paving the way for a sustainable energy future. This research sets a precedent for future studies that explore models incorporating variable weather and complex aerodynamics.

1.2 Motivation

The shift towards renewable energy sources is crucial to mitigate the adverse effects of climate change and reduce dependence on fossil fuels. Traditional wind turbines have been instrumental in this shift, but come with limitations such as high construction costs, visual and noise pollution, and the need for specific geographical locations. AWES present a revolutionary alternative that addresses these limitations and offers enhanced efficiency and flexibility.

Figure 1.1: UPWIND [1] kite model prototype.

AWES, as illustrated in Figure 1.1, utilize tethered kites or drones that can access higher altitudes where wind speeds are more consistent and stronger compared to those available to traditional wind turbines. This higher altitude wind harnessing capability results in a more reliable and potent energy source, as illustrated in Figure1.2. Additionally, the deployment of AWES involves less material and lower costs as they do not require massive support structures like conventional wind turbines.

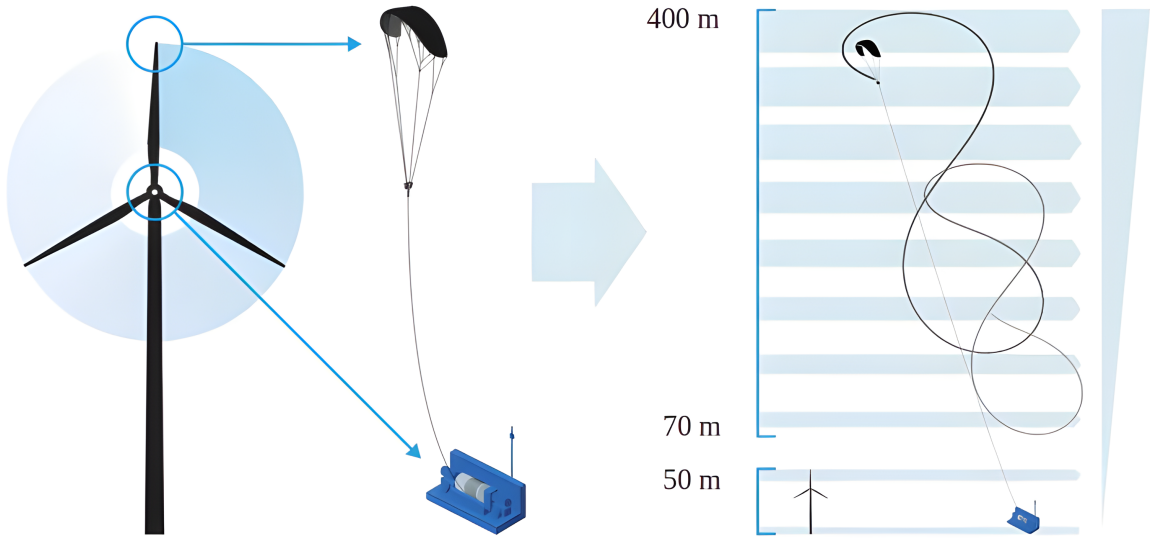

Figure 1.2: Fundamental idea behind Airborne Wind Energy systems with a ground-based generator [8].

The image demonstrates the fundamental idea behind AWES with a ground-based generator. Unlike traditional wind turbines that convert wind energy through large, rigid blades, AWES employs flexible kites or wings that are attached to a ground-based generator. As the kite moves with the wind, it pulls out the tether, turning the generator on to produce electricity. This method allows for efficient energy capture even in varying wind conditions.

1.3 Objectives

The primary objective of this dissertation, titled *Optimization of Energy within the Complete Cycle of Airborne Wind Energy Systems*, is to improve the efficiency and effectiveness of AWES through advanced simulations and optimized control strategies. The specific objectives are as follows.

- **Study of the Kite Dynamical Model in AWES** – Developed and implemented a dynamical model for the kite, capturing key forces and interactions during operation. This model was critical for understanding the kite's behavior during both the reel-out (traction) and reel-in (retraction) phases.
- **Simulation of Kite Dynamics and Control** – Built and refined a simulation platform using the AWEBox framework, enabling both 2D and 3D simulations. These simulations focused on analyzing and optimizing the kite's trajectory to ensure stability and efficiency throughout its full operational cycle.
- **Optimization of Kite Trajectories and Energy Output** – Successfully optimized kite trajectories for maximizing energy extraction during the reel-out phase and minimizing retraction time.

Algorithms were developed to fine-tune control variables such as reel-out speed and angle of attack.

- **Integration of Previous Research** – Extended previous research from the UPWIND project at FEUP and incorporated tools and findings from prior dissertations, adapting them to this study's specific objectives and simulation requirements.

The successful completion of these objectives is expected on bringing new insights on control strategies that optimize energy extraction efficiency.

1.4 Document Structure

This dissertation is structured as follows:

- **Chapter 1: Introduction** - Provides the context, objectives, and the structure of the dissertation.
- **Chapter 2: State of the Art in AWES and Kite Technologies** - Reviews existing technologies and principles of high wind power extraction, including traditional wind turbines and AWES.
- **Chapter 3: Dynamic Model of the Kite Power System** - Details the development of the dynamical model of a kite used in AWES.
- **Chapter 4: Simulation, Integration and Optimization of Kite Dynamics** - Describes the creation of the simulation platform and the implementation of 2D and 3D simulations.
- **Chapter 5: Simple Closed-Loop Trajectory** - Discusses the development and implementation of algorithms for optimizing kite trajectories.
- **Chapter 6: Closed-Loop Trajectory with Ascending and Retracting Phase** - Covers the integration of control strategies into a cohesive system and the validation of the system's performance through simulations and real-world tests.
- **Chapter 7: Conclusions and Future Work** - Summarizes the key findings of the dissertation, its contributions to the field, and suggests directions for future research.

This structure ensures a comprehensive exploration of the optimization of energy within the complete cycle of AWES, from theoretical foundations to practical implementations and evaluations.

Chapter 2

State of the Art in AWES and Kite Technologies

AWES operate by deploying an airborne device attached to a ground station. The airborne device captures wind energy and transmits it to the ground mechanically or electrically. There are two primary types of AWES based on how energy is generated: Ground-Gen (GG-AWES) and Fly-Gen (FG-AWES).

The key advantages [9] of AWES include:

1. **Higher Efficiency**: By operating at higher altitudes, AWES can capture stronger and more consistent winds, leading to increased energy production.

2. **Lower Environmental Impact**: The visual and noise pollution is significantly reduced as the infrastructure is less obtrusive and quieter compared to traditional turbines.

3. **Cost-Effectiveness**: The reduced need for large support structures and the flexibility in deployment locations make AWES more cost-effective.

4. **Scalability**: AWES can be scaled easily by increasing the number of kites or drones, providing a versatile solution for various energy demands.

As the world strives to achieve more aggressive climate goals, innovations like AWES are crucial. They not only enhance our ability to harness renewable energy, but also pave the way for a more sustainable and resilient energy infrastructure. This dissertation focuses on optimizing the complete energy cycle within AWES, with the aim of improving system efficiency and economic viability through advanced modeling, simulation, and control strategies.

2.1 Overview of AWES Configurations

Airborne Wind Energy Systems (AWES) can be classified into three primary configurations: Ground-Gen, Fly-Gen, and Hybrid systems. These configurations differ based on how power is generated and transmitted, each offering distinct approaches to harnessing high-altitude wind energy. The subsequent subsections will explore the specific characteristics and operational principles of each system.

2.1.1 Ground-Gen AWES

In GG-AWES [10], the airborne device generates mechanical energy by pulling the cord, which is connected to a generator on the ground. As the device flies in crosswind patterns, it maximizes the apparent wind speed, thereby generating significant tension in the tether. This tension drives a ground-based generator to produce electricity. The cycle includes a generation phase, where the tether is unwound, and a recovery phase, where the tether is rewound with minimal energy consumption.

GG-AWES are advantageous because of their simplicity and the fact that the heavy components (generators and other electrical equipment) remain on the ground. This makes the airborne component lighter and potentially more efficient in capturing wind energy. However, the intermittent nature of the generation and recovery phases can pose challenges in maintaining a continuous power output.

This technology can be further divided into several subtypes on the basis of their design and operational principles. For example, some systems use a single large kite, while others employ multiple smaller kites to increase efficiency and reliability. The design of the ground station, including the type of generator and the method of energy conversion, also varies among different GG-AWES implementations.

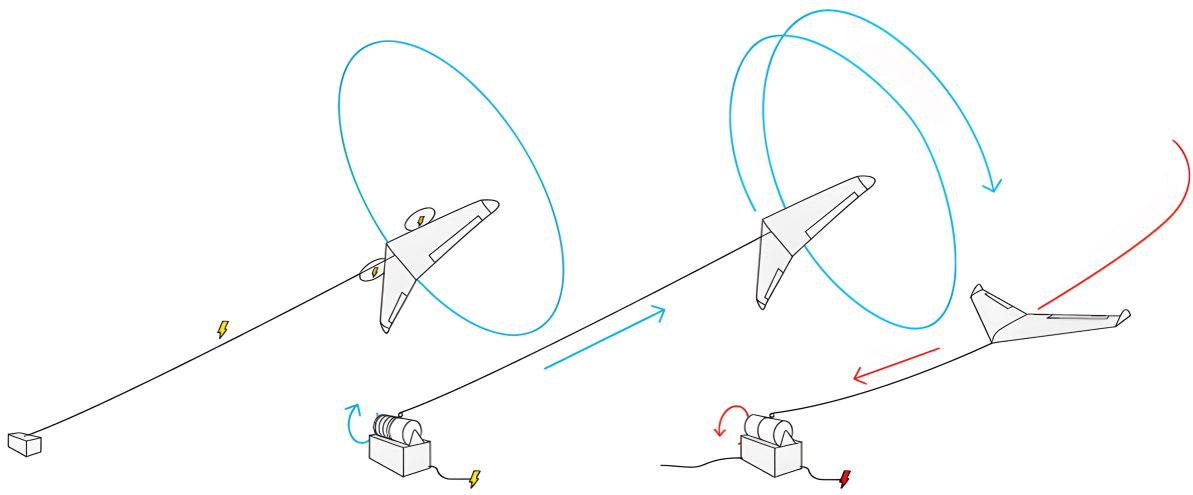

Figure 2.1: Schematic of Ground-Gen AWES [11].

2.1.2 Fly-Gen AWES

In FG-AWES [12], the airborne device itself carries generators and converts mechanical energy into electrical energy at altitude. This electricity is then transmitted to the ground via conductive tethers. FG-AWES can produce continuous power without the need for a recovery phase, but they require more complex and heavier airborne equipment.

FG-AWES are capable of generating electricity continuously as they do not need to retract the tether periodically. The major challenge with this system is the added weight and complexity of carrying electrical generators and other components aloft. Moreover, the conductive tether needs to be robust enough to transmit electricity while withstanding the mechanical stresses imposed by the wind and the motion of the airborne device.

These systems often resemble small aircraft or drones equipped with wind turbines. These systems must be carefully designed to balance the need for lightweight construction with the need for durability and reliability. Advances in materials science, particularly the development of lightweight composites and high-strength conductive materials, are crucial to the success of FG-AWES.

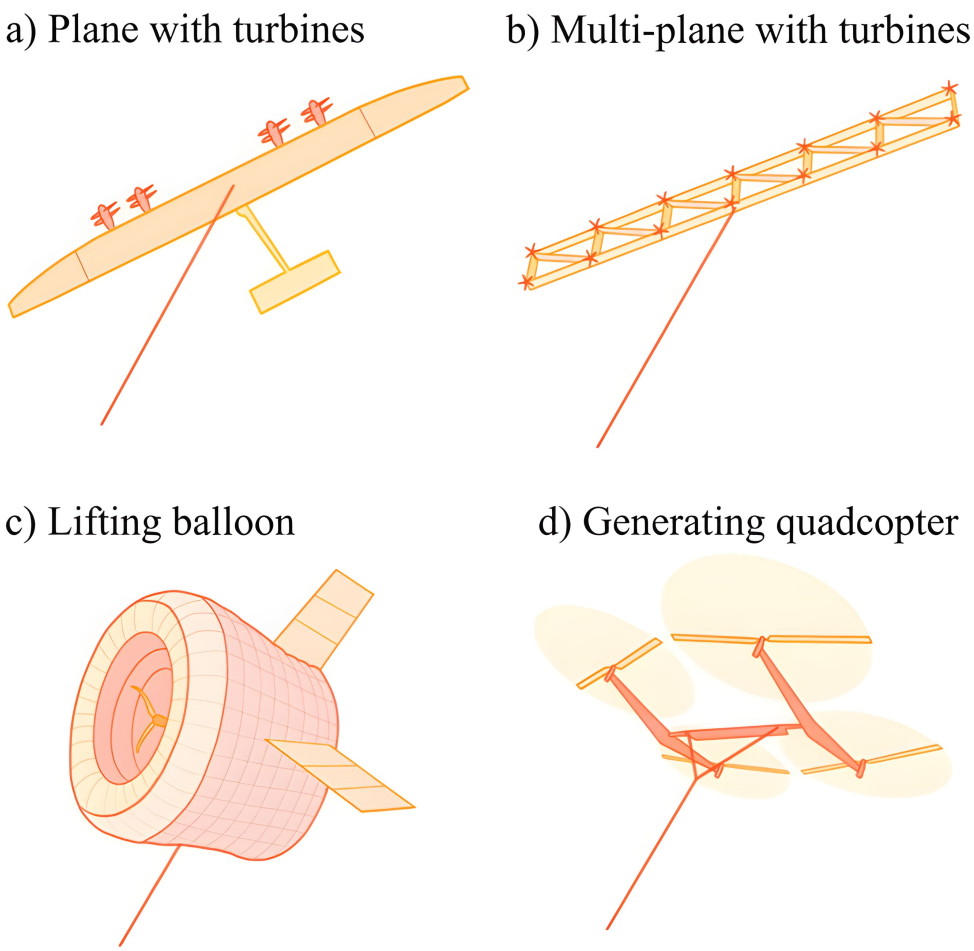

Figure 2.2: Schematic of Fly-Gen AWES [13].

2.1.3 Hybrid AWES

Hybrid AWES [14] combine elements of both GG-AWES and FG-AWES to take advantage of the benefits of each approach. For example, some hybrid systems use an airborne device that generates mechanical power, which is then converted to electrical power at both the altitude and the ground station. This allows continuous power generation, while also taking advantage of the lighter airborne components typical of GG-AWES.

Hybrid systems can be particularly advantageous in offshore applications, where the complexity and cost of deploying traditional wind turbines are higher. By combining the continuous power generation capability of FG-AWES with the simplicity and lower weight of GG-AWES, hybrid systems can offer a flexible and scalable solution for various deployment scenarios.

2.2 Challenges and Technological Developments

Despite their potential, AWES face several challenges that need to be addressed to achieve commercial viability [15].

1. **Durability and Reliability**: AWES must withstand harsh weather conditions and continuous mechanical stress on the tethers and airborne devices. Advances in materials science are crucial for improving the durability and longevity of these systems. The development of high-strength, lightweight materials for tethers and kite structures is essential for ensuring the operational reliability of AWES over extended periods.

2. **Control and Stability**: Maintaining stable flight and optimal control of the airborne device is critical for efficient energy generation. Research is focused on developing sophisticated control algorithms that can adapt to varying wind conditions and ensure the safe operation of the system. Control strategies involve complex dynamics and require real-time adjustments to the kite's position, tether length, and angle of attack to maximize power output while maintaining stability.

3. **Regulatory and Safety Issues**: Integrating AWES into existing airspace and ensuring their safe operation in proximity to populated areas and other infrastructure presents regulatory challenges. Collaboration with aviation authorities and the development of comprehensive safety protocols are necessary. Safety concerns include the potential for kite crashes, entanglement with other aerial vehicles, and interference with air traffic. Establishing clear regulatory frameworks and safety standards is crucial for the widespread adoption of AWES.

2.3 Current Research and Future Directions

Recent studies and pilot projects have demonstrated the feasibility of AWES [16]. Notable companies and research institutions are actively working to improve the technology and address the aforementioned challenges.

2.4 Case Study: UPWIND Project

The UPWIND project at the University of Porto focuses on developing efficient airborne wind energy technologies [1]. Key areas of research include automatic take-off and landing, optimization, and control of AWES, and multiple kite systems. The project aims to make airborne wind energy more efficient, cost-effective, and accessible. The UPWIND team has made significant contributions in modeling, path planning, and control strategies to maximize power output and ensure system stability.

The UPWIND project also explores the integration of AWES with existing energy grids, examining the economic and environmental benefits of such systems. By addressing both technical and practical aspects of the implementation of AWES, the UPWIND project aims to pave the way for the commercialization of airborne wind energy technologies.

The UPWIND project covers several critical research areas:

- **Automatic Take-off and Landing:** Developing new methods for circular take-off and landing (CTOL) and innovative bridle systems within the kite to enhance the efficiency and safety of these operations.
- **Identification, Modeling, and Estimation of AWES:** Providing detailed modeling and identification of kite generator systems (KGS) and applying moving horizon estimation techniques to improve the accuracy and reliability of these models.
- **Optimization and Control of AWES:** Implementing advanced control techniques, such as Model Predictive Control (MPC), and using adaptive mesh refinement techniques to solve the optimal control problem more efficiently. These methods ensure that the kite follows a predefined path that maximizes power output and maintains system stability.
- **Multiple Kite Systems:** Ensuring the robustness of the overall system, including collision avoidance and optimal layout configurations for multiple kites operating within the same area.

The UPWIND project has also been recognized for its innovative contributions to the field. The project was distinguished as one of the top three technologies in the iUP25k 2024 Competition (UPorto Business Ideas Competition) and has received support from several research funding bodies, including FCT/MCTES (PIDDAC).

By integrating these advanced research areas, the UPWIND project aims to overcome the current challenges faced by AWES and facilitate their large-scale deployment. This comprehensive approach

not only addresses technical hurdles but also explores the economic viability and environmental benefits of airborne wind energy systems, significantly contributing to the global shift toward renewable energy sources.

2.4.1 Future Prospects

The future of AWES looks promising with ongoing advances in aerodynamics, materials science, and control systems. As technology matures, it is expected to complement traditional wind energy sources and contribute significantly to the global renewable energy mix. Future research directions include the development of autonomous flight systems that can optimize power generation with minimal human intervention, the use of advanced materials to improve durability and performance, and the scaling up of AWES to utility-scale power generation. Collaboration between academia, industry, and government agencies will be essential to overcome the remaining challenges and accelerate the adoption of AWES [17].

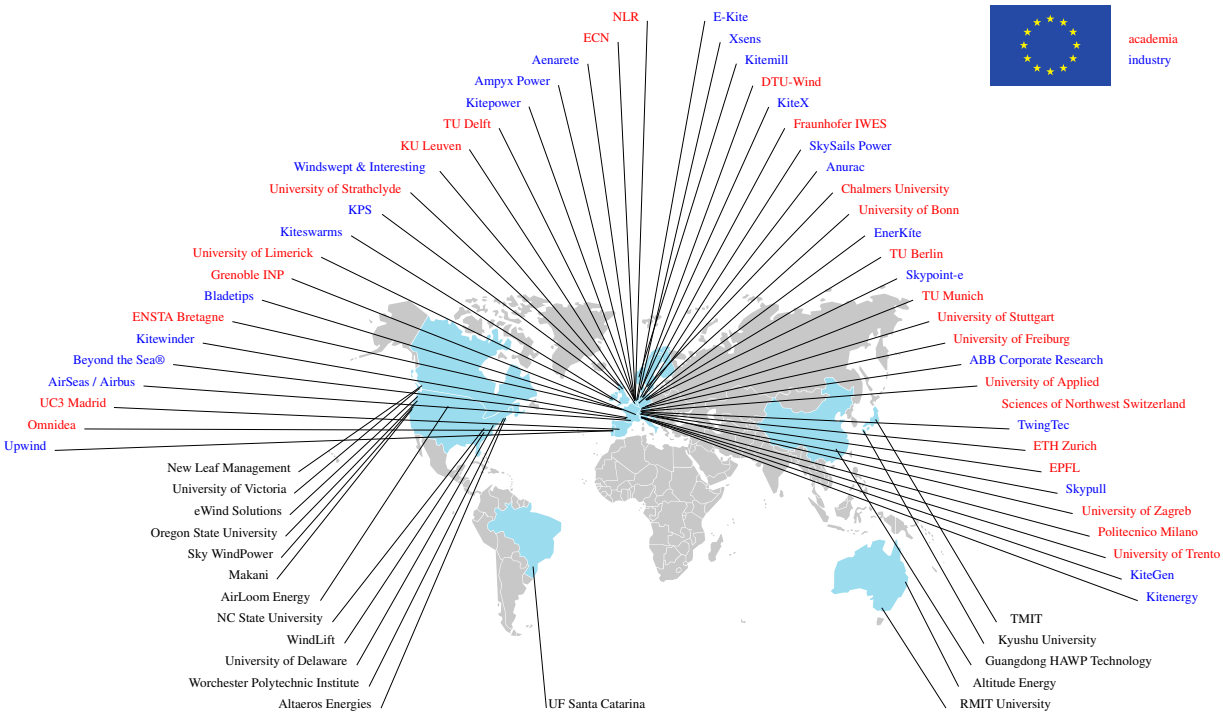

Figure 2.3: Institutions With Airborne Wind Energy Projects [18]

The global distribution of institutions that work on AWES projects, as shown in Figure 2.3, highlights the growing interest and investment in this technology. These institutions are involved in various aspects of AWES development, including aerodynamic modeling, control systems, and hardware design.

[19] provides an in-depth explanation of the different types of AWES and their potential applications, including tethered wings, kites, and other airborne devices. The report emphasizes the importance of optimizing both the design and operation of these systems to maximize energy yield and minimize environmental impact.

2.5 Conclusion

Airborne Wind Energy Systems represent a transformative approach to harnessing wind energy, leveraging high-altitude winds for more efficient and consistent power generation. Although challenges remain, continued research and technological advancements are paving the way for AWES to become a viable and impactful component of the renewable energy landscape.

By addressing key issues related to durability, control, and regulatory compliance, AWES has the potential to revolutionize the way we generate renewable energy. The successful integration of AWES into the global energy infrastructure will depend on the collaborative efforts of researchers, engineers, policymakers, and industry stakeholders.

Chapter 3

Dynamic Model of the Kite Power System

3.1 Introduction

This chapter delves into the mathematical and physical principles underlying the dynamic modeling of a Kite Power System (KPS). The model treats the kite as a point mass and incorporates various forces that influence its trajectory through the atmosphere. Understanding these dynamics is crucial for optimizing the kite's performance and efficiency.

3.2 Coordinate Systems

Understanding the motion and control of the Kite Power System (KPS) requires the definition of several coordinate systems, each tailored to represent different aspects of the kite's dynamics relative to the Earth, the air around it, and its own structure. These systems are essential for accurately describing the kite's orientation, position, and movement through space [20].

3.2.1 Global Coordinate System (G)

The global coordinate system is an inertial Cartesian system, which is fixed relative to the Earth. It is defined as follows:

- Origin: Located at the ground station of the KPS, providing a reference point for the system.
- Axes: The x-axis is aligned with the prevailing wind direction, ensuring that the wind velocity vector can be straightforwardly represented. The y-axis is orthogonal to the x-axis on the horizontal plane, and the z-axis is vertical, pointing upward.

This coordinate system serves as the reference for evaluating the kite's position and the direction of environmental forces such as wind.

3.2.2 Local Coordinate System (L)

The local coordinate system is a non-inertial spherical system centered at the kite defined by $(\vec{e}_r, \vec{e}_\phi, \vec{e}_\beta)$, which moves with the kite and is crucial for understanding its immediate environment:

- Origin: Positioned at the kite's center of mass, it is a noninertial reference frame because of the kite's motion.
- Axes: Defined by radial ($\vec{\mathbf{e}}_r$), azimuthal ($\vec{\mathbf{e}}_\phi$), and polar ($\vec{\mathbf{e}}_\beta$) unit vectors, which describe the orientation and position of the kite relative to the ground station.

The position of the kite p in this system is represented as:

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos(\beta)\cos(\phi) \\ r\cos(\beta)\sin(\phi) \\ r\sin(\beta) \end{bmatrix} \tag{3.1}$$

The rotation matrix from the local coordinate system to the global coordinate system is as follows:

$$R_{LG} = \begin{bmatrix} \cos(\beta)\cos(\phi) & -\sin(\phi) & -\sin(\beta)\cos(\phi) \\ \cos(\beta)\sin(\phi) & \cos(\phi) & -\sin(\beta)\sin(\phi) \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{3.2}$$

This orthogonal matrix can be transposed to convert from the global to the local coordinate system.

This system is particularly useful for analyzing the forces acting on the kite and its dynamic response to control inputs.

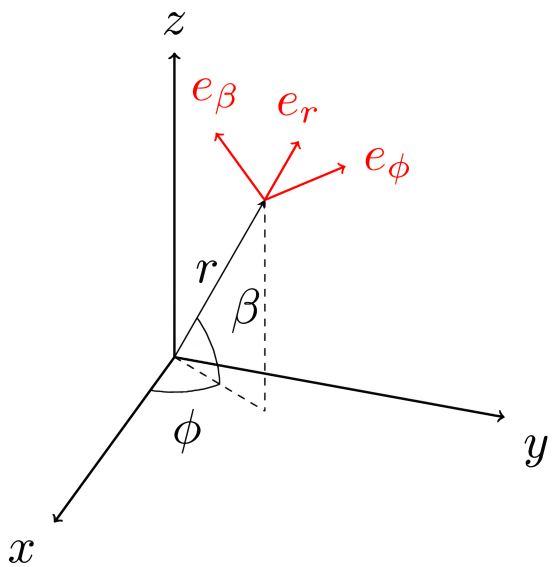

Figure 3.1: Global and Local Coordinate Systems - Referential Representation.

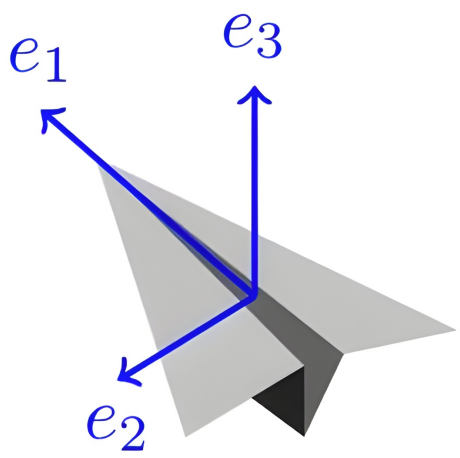

Figure 3.2: Global and Local Coordinate Systems - Kite Representation.

3.2.3 Body Coordinate System (B)

Aligned with the kite structure, the body coordinate system rotates and translates with the kite; it is defined by $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$, where $\vec{e}_1$ corresponds to the longitudinal axis, $\vec{e}_2$ to the lateral axis, and $\vec{e}_3$ to the vertical axis of the kite. With this coordinate system, we can analyze aerodynamic forces and control actions effectively:

- Origin: The coordinates coincide with the kite's center of mass.
- Axes: The $\vec{\mathbf{e}}_1$ axis aligns with the longitudinal axis of the kite, pointing from the tail to the head. The $\vec{\mathbf{e}}_2$ axis is orthogonal to $\vec{\mathbf{e}}_1$, extending horizontally towards the tip of the left wing. The $\vec{\mathbf{e}}_3$ axis is orthogonal to both, pointing vertically upward when the kite is in a neutral position.

This coordinate system is essential for implementing and understanding the control algorithms that influence the kite's orientation and flight path.

3.2.4 Coordinate Transformations

To facilitate the analysis and control of the kite, transformations between these coordinate systems are used. These transformations allow the conversion of vector quantities from one coordinate system to another, enabling comprehensive dynamic modeling and control system design.

The mathematical representation of these coordinate transformations is fundamental for deriving the equations of motion and for simulating the kite behavior under various operational conditions. Accurate coordinate transformations ensure that the control strategies developed are effective in real-world conditions.

Considering the position **p** that describes the mass point of the kite:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos(\beta)\cos(\phi) \\ r\cos(\beta)\sin(\phi) \\ r\sin(\beta) \end{bmatrix} \tag{3.3}$$

The rotation matrix from the L coordinate system to G is

$$\mathrm{R_{LG}} = \begin{bmatrix} \vec{\mathbf{e}}_r & \vec{\mathbf{e}}_\phi & \vec{\mathbf{e}}_\beta \end{bmatrix} = \begin{bmatrix} \cos(\beta)\cos(\phi) & -\sin(\phi) & -\sin(\beta)\cos(\phi) \\ \cos(\beta)\sin(\phi) & \cos(\phi) & -\sin(\beta)\sin(\phi) \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{3.4}$$

The rotation matrix from the G coordinate system to L is found to be $\mathrm{R_{GL}} = \mathrm{R_{LG}^{-1}} = \mathrm{R_{LG}^{\top}}$.

Considering the apparent wind velocity $\mathbf{v}_a = \mathbf{v}_w - \dot{\mathbf{p}}$. We assume that its radial component $\mathbf{v}_{a,r}$ is strictly positive and that the longitudinal axis of the kite body aligns naturally with the apparent wind velocity, that is $\vec{\mathbf{e}}_1 = -\mathbf{v}_a / \|\mathbf{v}_a\|$.

Considering ψ the roll angle that measures rotation around the longitudinal axis $\left(\vec{\mathbf{e}}_1\right)$ and considering that $\tilde{e}_2 = \vec{\mathbf{e}}_2$ (for $\psi = 0$) is initially in the plane τ, tangent to a sphere centered at the origin (containing the axis $\vec{\mathbf{e}}_\phi$ and $\vec{\mathbf{e}}_\beta$), we have that $\tilde{e}_2 \perp \vec{\mathbf{e}}_r$, and $\tilde{e}_2 \perp \vec{\mathbf{e}}_1$. Thus, we can define $\tilde{e}_2 = \frac{\vec{\mathbf{e}}_r \times \vec{\mathbf{e}}_1}{\|\vec{\mathbf{e}}_r \times \vec{\mathbf{e}}_1\|}$.

3.3 Acting Forces

Understanding the dynamics of the Kite Power System (KPS) requires a comprehensive analysis of the forces that act on the kite. These forces determine the trajectory and stability of the kite and are essential for developing effective control strategies.

3.3.1 Aerodynamic Forces

Aerodynamic forces are critical in the operation of kites and include lift and drag, which are influenced by the kite's velocity relative to the air, its shape, and its angle of attack:

- **Lift (L):** Acts perpendicular to the direction of the apparent wind velocity. It is primarily a function of kite speed, angle of attack, and air density. Mathematically, the lift can be expressed as:

$$L = \frac{1}{2}\rho v^2 A c_L(\alpha), \tag{3.5}$$

  where ρ is the air density, v is the relative wind speed, A is the kite surface area, c_L is the lift coefficient and α is the angle of attack.

- **Drag (D):** Acts opposite to the direction of the apparent wind velocity and increases with the speed of the kite. It is calculated by:

$$D = \frac{1}{2}\rho v^2 A c_D(\alpha), \tag{3.6}$$

where c_D is the drag coefficient.

Aerodynamic forces $\vec{F}_{aer}$ include lift and drag forces, which depend on the apparent wind velocity and the angle of attack α:

$$\vec{F}_{aer}(\alpha) = \frac{1}{2}\rho A \|v_a\|^2 (c_L(\alpha)\vec{e}_3 - c_D(\alpha)\vec{e}_1) \tag{3.7}$$

where ρ is the density of the air, A is the area of the wing, c_L is the coefficient of lift and c_D is the coefficient of drag. These forces are vital for maintaining flight and maneuvering the kite, especially when adjusting the angle of attack to optimize performance.

3.3.2 Gravitational Force

The gravitational force acts constantly towards the center of the Earth and affects the kite's ability to maintain altitude:

$$F_{\text{grav}} = mg, \tag{3.8}$$

where m is the mass of the kite and g is the acceleration due to gravity. This force must be balanced by the lift generated by the kite to maintain flight.

The gravitational force $\vec{F}_{grav}$ acts downward and is given by:

$$\vec{F}_{grav} = -mg\vec{e}_z = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}_G = \begin{bmatrix} -mg\sin(\beta) \\ 0 \\ -mg\cos(\beta) \end{bmatrix}_L \tag{3.9}$$

3.3.3 Tether Tension

The tether connects the kite to the ground station and transmits mechanical energy. The tension in the tether arises from the combination of aerodynamic forces and the kite's weight:

$$T = \sqrt{L^2 + D^2} - F_{\text{grav}}\sin(\theta), \tag{3.10}$$

where θ is the angle of the tether relative to the horizontal plane. This tension is crucial for the energy generation phase, as it influences the power output by affecting the kite's position and the wind area it can harness.

The tension force of the tether $\vec{F}_{th}$ acts along the tether and is expressed as:

$$\vec{F}_{th} = -T\vec{e}_r = \begin{bmatrix} -T \\ 0 \\ 0 \end{bmatrix}_L \tag{3.11}$$

3.3.4 Mathematical Integration of Forces

The total force acting on the kite can be expressed as the vector sum of the aerodynamic, gravitational, and tension forces of the tether. This integration is vital for deriving the equations of motion that dictate the kite's trajectory:

$$\mathbf{F}_{\text{total}} = \mathbf{F}_{\text{aer}} + \mathbf{F}_{\text{grav}} + \mathbf{F}_{\text{tether}}, \tag{3.12}$$

where $\mathbf{F}_{\text{aer}}$ represents the vector sum of lift and drag, $\mathbf{F}_{\text{grav}}$ is directed downward, and $\mathbf{F}_{\text{tether}}$ acts along the line of the tether.

Understanding these forces and their interactions is crucial for modeling the kite's behavior, designing its control systems, and optimizing its energy production efficiency. This understanding forms the foundation for the dynamic simulations and real-world applications of KPS.

3.4 Dynamic Equations

The motion of the kite is described through Newton's laws, applied in the spherical coordinate framework. The state variables, representing the kite's dynamic state, are defined as follows:

$$\mathbf{x} = (r, \phi, \beta, \dot{r}, \dot{\phi}, \dot{\beta}), \tag{3.13}$$

where each component represents the radial distance, azimuthal angle, elevation angle, and their respective rates of change.

Assuming that the radial acceleration can be directly controlled by a_t, we can define the state variables $\mathbf{x} = (r, \phi, \beta, \dot{r}, \dot{\phi}, \dot{\beta})$, the control variables $\mathbf{u} = (a_t, \alpha, \psi)$ and the dynamic equation of the system as:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) = \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} r \\ \phi \\ \beta \\ \dot{r} \\ \dot{\phi} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \dot{\phi} \\ \dot{\beta} \\ a_t \\ \frac{1}{mr\cos(\beta)} \mathrm{F}_\phi \\ \frac{1}{mr} \mathrm{F}_\beta \end{bmatrix} \tag{3.14}$$

The dynamic equations, derived from the force balance in each coordinate direction, are the following.

$$\ddot{r} = \frac{F_{\text{th}} + F_{\text{aer}}(\alpha)\cos(\beta) - mg\sin(\beta)}{m}, \tag{3.15}$$

$$r\ddot{\phi} = \frac{F_\phi}{m}, \tag{3.16}$$

$$r\cos(\beta)\ddot{\beta} = \frac{F_\beta - mg\cos(\phi)\sin(\beta)}{m}, \tag{3.17}$$

where $F_{\text{th}}, F_{\text{aer}}(\alpha), F_{\phi}$, and F_{β} are the tether, aerodynamic, and component forces acting along the respective coordinate axes.

The kite motion is governed by the following dynamic equation:

$$m\ddot{p} = \vec{F}_{th} + \vec{F}_{grav} + \vec{F}_{aer}(\alpha) + \vec{F}_{inert} \tag{3.18}$$

where $\vec{F}_{inert}$ represents inertial forces (centrifugal and Coriolis forces). In the local coordinate system, this can be expanded as:

$$\ddot{p} = \begin{bmatrix} \ddot{r} \\ r\ddot{\phi}\cos(\beta) \\ r\ddot{\beta} \end{bmatrix}_L + \begin{bmatrix} -r\dot{\beta}^2 - r\dot{\phi}^2\cos^2(\beta) \\ 2\dot{r}\dot{\phi}\cos(\beta) - 2r\dot{\phi}\dot{\beta}\sin(\beta) \\ 2\dot{r}\dot{\beta} + r\dot{\phi}^2\cos(\beta)\sin(\beta) \end{bmatrix}_L - \frac{1}{m}\vec{F}_{inert} \tag{3.19}$$

These equations form the basis for simulations that predict the kite's performance, allowing for detailed scenario analysis and system optimization.

3.5 Control Variables

Effective control of the kite is achieved by manipulating several variables, including tether acceleration, angle of attack, and roll angle.

3.5.1 Tether Acceleration

The tether acceleration a_t controls the reel-in and reel-out speed of the tether. The maximum power output is achieved when the kite's reel-out speed is approximately one-third of the wind speed.

$$v_{reel_out} \approx \frac{1}{3}v_w \tag{3.20}$$

During the recovery phase, a balance must be found between the reel-in speed and energy consumption to optimize efficiency.

3.5.2 Angle of Attack

The angle of attack α is the angle between the kite's wing chord and the apparent wind velocity. The control of α is crucial for maximizing lift during the production phase and minimizing lift during the recovery phase.

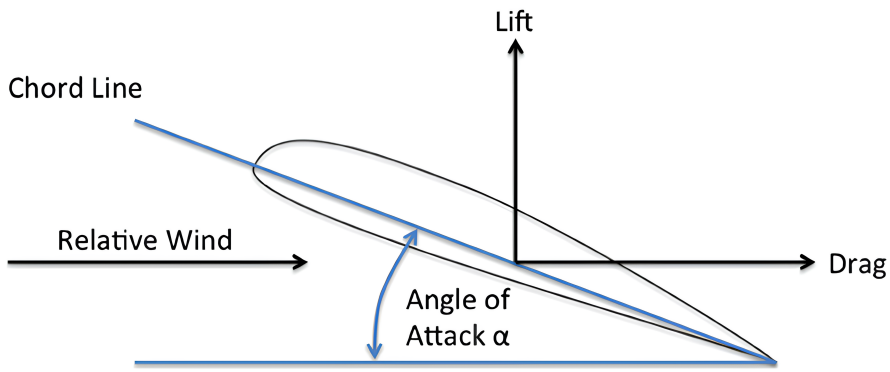

Figure 3.3: Angle of Attack [21].

3.5.3 Roll Angle

The roll angle ψ measures the rotation around the longitudinal axis of the kite $\vec{e}_1$. This angle is controlled to direct the kite's heading and stabilize its flight.

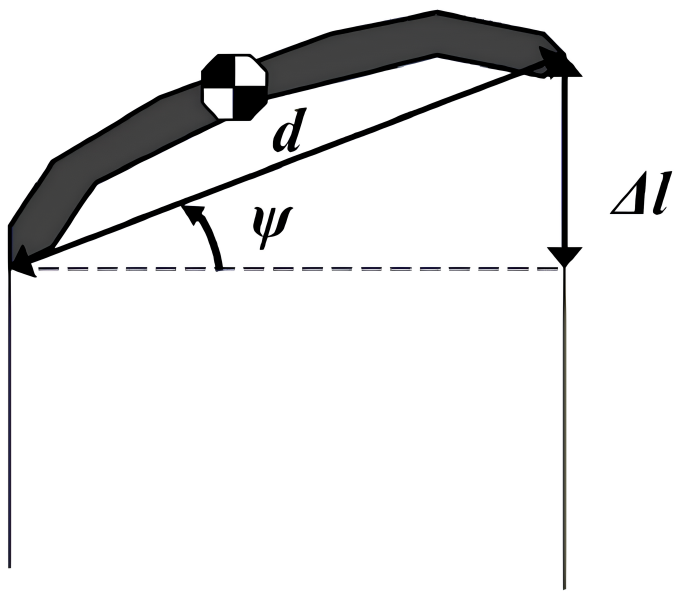

Figure 3.4: Roll angle control mechanism [21].

3.6 Simulation and Control

Using the established dynamic model, various control strategies are implemented to optimize the power output and operational efficiency:

- **Tether Length Control:** Regulating the tether length to optimize the flight path and maximize energy capture.
- **Angle of Attack Control:** Adjusting the angle of attack of the kite to balance the lift and drag, optimizing the speed and trajectory for the prevailing wind conditions.
- **Flight Path Control:** Utilizing advanced algorithms to determine optimal flight paths that enhance energy generation.
- **Winch Control:** Managing the wind speed to efficiently convert mechanical energy during the power generation and retraction phases.

The dynamic model forms the foundation for implementing these control strategies using simulations.

Chapter 4

Simulation, Integration and Optimization of Kite Dynamics

4.1 Introduction

This chapter provides a detailed description of the dynamic model of a PKG system, which is essential for optimizing energy extraction from high-altitude winds. The dynamic modeling, simulation, and control strategies discussed here form the foundation for advanced simulations and real-world applications of AWES.

4.2 Optimal Control

Optimal control is a sophisticated mathematical framework designed to determine the best control inputs for a system to achieve a desired outcome [22]. It is a crucial tool in various fields, including engineering, economics, robotics, and more, where decision-making is critical over time. In the realm of Airborne Wind Energy Systems (AWES), optimal control is instrumental in maximizing energy capture while minimizing operational costs, thus ensuring efficient and sustainable energy production.

The essence of optimal control lies in identifying a control policy that guides the system from an initial state to a desired final state, optimizing a predefined objective function along the way. This objective function, often referred to as a cost or performance index, quantifies the goal to be achieved, such as minimizing energy consumption or maximizing energy output. In the context of AWES, typical objective functions may include the total energy harvested during a cycle or the overall efficiency of energy conversion processes.

An optimal control problem typically consists of several key components:

- **Dynamic Model of the System**: This involves a set of differential equations describing the evolution of the system's state variables over time. These equations encapsulate the physics and constraints governing the system's behavior.

- **Control Variables**: These are the decision inputs that influence the system's state trajectory. In AWES, control variables could include parameters like tether tension, reel-out speed, and angles of attack.
- **Objective Function**: The function to be optimized, representing the desired performance metric. This could be a measure of energy efficiency, total power output, or other relevant criteria.
- **Constraints**: These include initial and final state constraints, control constraints, and possibly state constraints. For AWES, constraints might cover physical limitations of the kite, safety regulations, and environmental factors such as wind conditions.

To solve the optimal control problems in this thesis, a direct method is employed. In optimal control, there are two main approaches: direct and indirect methods. Indirect methods rely on Pontryagin's Maximum Principle (PMP), requiring the solution of a boundary value problem (BVP) derived from the Hamiltonian. This approach is computationally complex, particularly for large, nonlinear systems like AWES.

In contrast, direct methods discretize the control problem, transforming it into a nonlinear programming problem (NLP). This allows the use of numerical optimization techniques, making direct methods more practical for high-dimensional systems. In this thesis, the problem is discretized and solved as an NLP using the CASADi framework, which efficiently computes gradients, Jacobians, and Hessians. The IPOPT solver, integrated with CASADi, is used for solving the NLP, offering robust handling of large-scale problems.

The AWEBox library and HSL solvers model the complex dynamics of AWES, enabling detailed simulations and trajectory optimization. The focus is on maximizing energy capture and minimizing losses throughout the kite's flight cycle, improving the efficiency of AWES, as outlined in previous chapters.

4.3 Discretization Methods

The computational solution of optimal control problems necessitates discretization, which transforms continuous problems into discrete ones for optimization solvers. As described in chapter 4 of [23], various discretization methods can be applied. Among the simplest and most effective are the following:

- **Explicit Euler Method:**

$$z_N(t_{i+1}) := z_N(t_i) + h_i f(t_i, z_N(t_i)) \tag{4.1}$$

- **Implicit Euler Method:**

$$z_N(t_{i+1}) := z_N(t_i) + h_i f(t_{i+1}, z_N(t_{i+1})) \tag{4.2}$$

- **Taylor Order 2 Method:**

$$z_N(t_{i+1}) := z_N(t_i) + h_i f(t_{i+1}, z_N(t_{i+1})) + \frac{h_i^2}{2} f'(t_{i+1}, z_N(t_{i+1})) \tag{4.3}$$

- **Runge-Kutta Method:**

$$\begin{aligned} k_1 &= f(t_i, z_N(t_i)) \\ k_2 &= f\left(t_i + \frac{h_i}{2}, z_N(t_i) + \frac{h_i}{2} k_1\right) \\ k_3 &= f\left(t_i + \frac{h_i}{2}, z_N(t_i) + \frac{h_i}{2} k_2\right) \\ k_4 &= f\left(t_i + h_i, z_N(t_i) + h_i k_3\right) \\ z_N(t_{i+1}) &= z_N(t_i) + \frac{h_i}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \tag{4.4}$$

Each of these methods has its own advantages and trade-offs in terms of stability and accuracy, making them suitable for different types of optimal control problems. In this study, the Runge-Kutta method was specifically used for solving the simulation problems, due to its balance of accuracy and computational efficiency.

4.3.1 Computational Tools for Optimization

For this thesis, we focus on the direct method for solving optimal control problems, which involves discretizing the problem and solving it as a large-scale nonlinear programming (NLP) problem. This approach is computationally intensive, but offers a robust framework for complex systems such as AWES. Several optimization solvers are available, including KNITRO [24] and IPOPT [2], widely recognized in the literature for their efficacy in handling optimal control problems. More recent tools such as WORHP [25] are also gaining attention, as shown in Figure 4.1.

MATLAB is a commonly used interface for these solvers, providing environments like ICLOCS (Imperial College London Optimal Control Solver) [27] and AMPL (A Mathematical Programming Language) [28]. ICLOCS allows users to input dynamic equations directly, handling the discretization internally, while AMPL requires pre-discretized input, giving the user control over the discretization method.

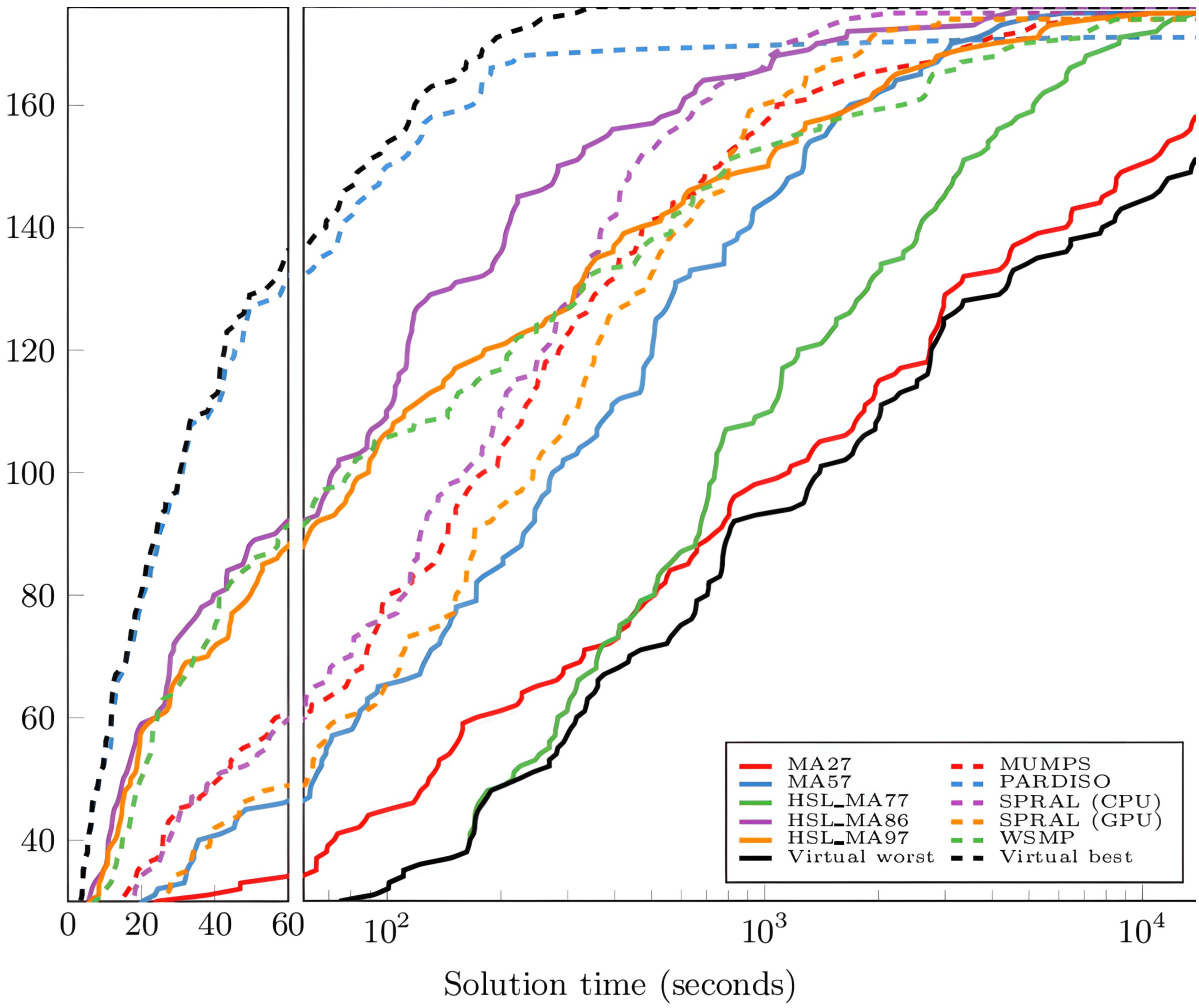

Figure 4.1: Comparison of Ipopt performance over various linear solvers using the two-dimensional partial differential equation test problem set. [26]

4.4 Numerical Methods and Solvers

Numerical methods play a crucial role in the optimization of complex systems, especially in the context of Airborne Wind Energy Systems (AWES). In this research, the direct method is employed to discretize the optimal control problem. This involves converting the continuous problem into a finite-dimensional nonlinear programming problem (NLP) that can be solved using numerical optimization techniques.

4.4.1 Discretization and Implementation

The direct method discretizes both the state and control variables over a series of time intervals. The system dynamics is approximated at these discrete points, allowing the continuous-time optimal control

problem to be transcribed into an NLP. The discretization technique used in this study involves a Zero-Order Hold (ZOH) approach, where control inputs are held constant within each interval. This approach simplifies the implementation of control and is particularly effective for systems with discrete control actions.

4.4.2 Optimization Solvers

To solve the resulting NLP, advanced optimization solvers are utilized. The primary solver used in this research is IPOPT (Interior Point OPTimizer), which is renowned for its efficiency in handling large-scale nonlinear problems. IPOPT uses a barrier method to deal with constraints and is particularly adept at handling the sparse structure typical of large optimization problems.

- **IPOPT:** IPOPT serves as the core solver for the optimization tasks. It employs an interior point method to find local solutions to the NLP, efficiently managing both equality and inequality constraints. The solver's ability to handle large and sparse systems makes it ideal for the complex optimization problems encountered in AWES.
- **HSL Solvers:** The Harwell Subroutine Library (HSL) solvers, such as MA57 and MUMPS, are integrated with IPOPT for solving the linear systems that arise during the optimization process. These solvers are specifically designed to handle sparse matrices, which are common in large-scale NLPs, thereby enhancing the computational efficiency.
- **CASADi:** CASADi is employed as a symbolic framework for the automatic differentiation and numeric optimization of the control problem. It facilitates the setup of the NLP by providing accurate computations of gradients, Jacobians, and Hessians, which are critical for the optimization algorithm's convergence and accuracy.
- **AWEBox Library:** The AWEBox library is utilized for modeling the AWES dynamics and integrating them with the optimization framework. It provides specialized functions and models tailored to airborne wind energy systems, enabling detailed simulations and optimizations.

The integration of these numerical methods and solvers allows for the effective resolution of the optimal control problem, ensuring accurate and efficient optimization. This methodology supports the research objective of enhancing the performance and efficiency of AWES, thus contributing to the advancement of renewable energy technologies.

4.5 Simulation, Integration, and Optimization of Kite Dynamics

The process of simulating, integrating, and optimizing the dynamics of the kite power system involves several critical stages. Each stage is designed to ensure accurate modeling of the kite's behavior under various conditions and optimize the control strategies for maximum efficiency and performance.

4.5.1 Initial Simulation with MATLAB and Simulink

The initial integration of the kite dynamics was carried out using MATLAB and Simulink. Simulink provided a robust platform for modeling and simulating the dynamic behavior of the kite, allowing the incorporation of various forces and control inputs. This environment was particularly useful for visualizing the response in the time domain of the system and verifying the accuracy of the dynamic equations.

In this stage, the dynamic equations of the kite were integrated into Simulink, capturing the complex interplay between aerodynamic forces, the tether dynamics, and control input. The Simulink graphical interface facilitated the construction of a detailed model, allowing simulation of different operational scenarios such as steady-state flight, turbulent wind conditions, and maneuvering operations. Despite its robustness for initial validation and exploration, Simulink was not optimized for solving complex optimal control problems, leading to the need for more specialized tools.

4.5.2 Optimization with CasADi and MATLAB

To address the optimal control problems associated with kite dynamics, CasADi and MATLAB were initially used. CasADi is an open source tool for non-linear optimization and algorithmic differentiation that provides efficient and flexible interfaces to define and solve optimization problems. The integration of CasADi with MATLAB allowed for the formulation of the kite's optimal control problem and the execution of simulations to find optimal trajectories.

The initial approach involved using an example model provided by my supervisors, which helped in understanding the CasADi integration process with MATLAB. This example model served as a starting point for developing the optimal control framework for kite dynamics. However, during this phase, it was observed that CasADi presented inconsistencies and performance issues depending on the specific versions of CasADi and MATLAB being used. These issues included solver instability, convergence problems, and discrepancies in the simulation results, necessitating a more reliable and robust solution for long-term research and development.

4.5.3 Transition to AWEBox and use of IPOPT with HSL Solvers

To address the limitations encountered with CasADi, the decision was made to transition to the AWEBox library. AWEBox is a comprehensive toolbox specifically designed for modeling and optimizing airborne wind energy systems. It incorporates CasADi, but also provides a structured framework and additional tools that enhance reliability and flexibility.

AWEBox offered several advantages over standalone CasADi:

- **Integrated Framework:** AWEBox provides a cohesive environment that simplifies the setup and execution of simulations and optimizations, reducing the potential for errors and inconsistencies.

- **Extended Functionality:** The library includes various utilities and predefined models tailored for airborne wind energy, streamlining the modeling process, and enhancing the accuracy of simulations.
- **Solver Flexibility:** AWEBox supports multiple solvers, including IPOPT (Interior Point OPTimizer) from the COIN-OR project. IPOPT, accessible through a license for students, is particularly effective for large-scale non-linear optimization problems, providing flexibility in choosing the most suitable solver for specific problems.

AWEBox is a Python toolbox for modeling and optimal control of multiple kite systems for airborne wind energy (AWE). It provides interfaces that aim to relieve the user of the burden of:

- Generating optimization-friendly system dynamics for different combinations of modeling options.
- Formulating optimal control problems for common multi-kite trajectory types.
- Solving the trajectory optimization problem reliably.
- Postprocessing and visualizing the solution and performing quality checks.
- Tracking MPC design and handling for offline closed-loop simulations.

The main focus of the toolbox is on rigid-wing, lift- and drag-mode multiple-kite systems. AWEBox runs on Python 3.8 and is heavily dependent on the modeling language CasADi, which is a symbolic framework for algorithmic differentiation. CasADi also provides an interface to the NLP solver IPOPT. It is optional but highly recommended to use HSL linear solvers as a plugin with IPOPT. In our case, we used these add-ons to enhance the accuracy of our trajectory simulations.

4.5.4 Environment Setup and Time Investment

The transition to AWEBox and the development of the necessary configuration files required significant time and effort. Ensuring that all configurations worked correctly involved extensive testing and debugging in a simulated environment using Python 3.8.19 on Ubuntu 24.04 LTS. This phase was particularly time-consuming as it involved addressing compatibility issues, solver integration problems, and ensuring the stability of the simulations. The use of a simulated Ubuntu environment provided a controlled and consistent setup, facilitating the troubleshooting process and ensuring that the developed solutions would be robust and reliable.

4.5.5 Simulation and Optimization Scenarios

With the custom model integrated into AWEBox, various simulation and optimization scenarios were executed. These included:

- **Closed-loop Trajectories:** Simulations of closed-loop flight paths, ensuring stable and efficient energy capture by maintaining optimal kite positions relative to wind conditions.
- **Point-to-Point Trajectories:** Optimization of trajectories from an initial point to a final point, maximizing energy efficiency and minimizing control effort. These simulations were critical for understanding the optimal paths for different wind conditions and operational constraints.
- **Dynamic Operations:** Simulations of dynamic operations such as take-off, landing, and mid-flight adjustments to external conditions. These scenarios tested the robustness of the control strategies and the system's ability to adapt to changing environmental conditions.

The use of AWEBox, with its robust and flexible framework, significantly improved the reliability and efficiency of simulations and optimizations. The ability to seamlessly integrate custom models and utilize advanced solvers like IPOPT provided a powerful platform to advance research on airborne wind energy systems. Furthermore, the extended functionality of AWEBox allowed for the exploration of various operational strategies and their impacts on system performance and energy yield.

Dependency	**Version**
Python	3.8.19
CasADi	3.6.4
cycler	0.11.0
fonttools	4.43.0
kiwisolver	1.4.4
matplotlib	3.7.5
numpy	1.24.4
packaging	21.3
Pillow	10.2.0
pyparsing	3.0.9
python-dateutil	2.8.2
scipy	1.10.1
six	1.16.0
tk	0.1.0
tabulate	0.8.10
Additional Requirements	
coinhsl	2022-11-09
metis	4.0.3

Table 4.1: AWEBox Required Dependencies and Hardware Recommendations

Table 4.1 presents all the requirements that AWEbox needs to operate properly. It is crucial to use the specific versions listed, as AWEbox relies on these exact versions to ensure proper synchronization of libraries. Additionally, it is recommended to use a machine with at least 32GB of RAM to ensure sufficient computational power for smooth operation. This amount of memory is necessary to handle the computational demands and data processing efficiently.

4.5.6 Advanced Control Strategies

The AWEBox framework utilized advanced linear solvers such as Coin-HSL [3] and MA57 [29] to solve the large-scale sparse systems of equations that arise during optimization. The use of these solvers was crucial for efficiently handling the computational demands of the optimization problems, particularly those involving high-dimensional state and control variables. Coin-HSL and MA57 provided robust and efficient solutions, significantly contributing to the overall stability and performance of the system.

4.6 Optimal Control and Optimization Objectives

In this research, the main focus is on optimizing the control strategies of the kite system, with trajectory optimization being a direct consequence of these strategies. The key objectives are:

- **Control Strategy Optimization:** Fine-tuning control variables such as tether tension, reel-out speed, and angle of attack to ensure stable and efficient operation. The trajectory of the kite is a result of these optimized control actions.
- **Power Output Maximization:** Maximizing the average power output over a complete pumping cycle, while minimizing energy losses during the retraction phase to improve overall system efficiency.

To achieve these objectives, we consider the optimal control problem for the time interval $t \in [0, t_f]$, where t is in seconds. The state vector is defined as $x = (r, \beta, \dot{\beta})$ and the control vector as $u = (v_t, \alpha)$. The problem can be formulated as follows.

4.6.1 Problem Formulation

The problem formulation for our simulations remains consistent between chapter 5 and 6. The primary structure and constraints of the model are the same across both chapters, focusing on the dynamics and control of the kite system. However, the key differences lie in the number of windings of the kite and the initial and final tether lengths. These parameters were adjusted to study their impact on system performance, energy capture, and overall efficiency.

Objective: Minimize the cost functional $J(x,u)$:

$$\text{Minimize} \quad J(x,u) = -\int_0^{t_f} \dot{r}T \, \mathrm{d}t \tag{4.5}$$

The total energy produced over the time period is:

$$E(t_f) = \int_0^{t_f} P(t) \, \mathrm{d}t \tag{4.6}$$

where the instant power production $P(t)$ is given by:

$$P(t) = -v_t T \tag{4.7}$$

In order to focus the energy production at the beginning of the time interval, we modify the objective function to:

$$\text{Minimize} \quad -\int_0^{t_f} e^{-t} \dot{r}T \, \mathrm{d}t \tag{4.8}$$

Dynamic Constraints:

$$\dot{r} = v_t \tag{4.9}$$

$$\ddot{\beta} = \frac{1}{mr}\left(-gm\cos(\beta) + F_{\text{aer},\beta}(\alpha) - 2m\dot{r}\dot{\beta}\right) \tag{4.10}$$

Input Constraints:

$$-v_w \leq v_t \leq v_w \quad \forall t \in [0, t_f] \tag{4.11}$$

$$-\alpha_{\max} \leq \alpha \leq \alpha_{\max} \quad \forall t \in [0, t_f] \tag{4.12}$$

End-Point Constraints:

$$x(t_0) = x_0 = (r_0, \beta_0, \dot{\beta}_0) \tag{4.13}$$

$$x(t_f) = x_f = (r_f, \beta_f, \dot{\beta}_f) \tag{4.14}$$

Bounded-State Constraints:

$$r_{\min} \leq r \leq r_{\max} \tag{4.15}$$

$$\beta_{\min} \leq \beta \leq \beta_{\max} \tag{4.16}$$

Tether Tension T is defined as:

$$T = -gm\sin(\beta) + F_{\text{aer},r}(\alpha) + mr\dot{\beta}^2 \tag{4.17}$$

where:

- $F_{\text{aer},\beta}$ and $F_{\text{aer},r}$ denote the aerodynamic forces in the elevation and radial directions, respectively.
- v_w is the wind speed.
- $\alpha_{\max}$ represents the maximum allowable angle of attack.
- $r_{\min}$ and $r_{\max}$ define the bounds for the radius.
- $\beta_{\min}$ and $\beta_{\max}$ set the limits for the elevation angle.

This optimal control formulation aims to determine the control inputs v_t and α that optimize the system's performance, ensuring maximum energy extraction while adhering to physical and operational constraints.

The aerodynamic coefficients are expressed as:

$$c_D(\alpha) = 0.012 + 0.01\alpha \tag{4.18}$$

$$c_L(\alpha) = 0.3 + 0.1\alpha \tag{4.19}$$

These coefficients and other parameters are detailed in Table 4.2.

Parameter	Value
ρ	1.2 kg m^{-3}
v_w	10 m s^{-1}
g	9.8 m s^{-2}
m	1 kg
A	10 m^2

Table 4.2: Parameters

4.6.2 System Constraints

To ensure safe and efficient operation of the kite power system, the following constraints are implemented. These limitations are critical to preventing mechanical failures and optimizing overall performance.

1. **Tether Force Limits**
   The tether force, F_{tether}, is restricted to maintain structural integrity:

$$50\,\text{kN} \leq F_{\text{tether}} \leq 1800\,\text{kN} \tag{4.20}$$

2. **Airspeed Bounds**
   To ensure adequate lift and prevent excessive aerodynamic forces, the airspeed, V_{airspeed}, is bounded as follows:

$$10\,\text{m/s} \leq V_{\text{airspeed}} \leq 32\,\text{m/s} \tag{4.21}$$

3. **Angle Constraints**
   The following angles are constrained to maintain aerodynamic stability:

   - **Angle of Attack (AoA), α:**

$$-6^\circ \leq \alpha \leq 9^\circ \tag{4.22}$$

   - **Side-Slip Angle, β:**

$$-20^\circ \leq \beta \leq 20^\circ \tag{4.23}$$

4. **Rotation Limits**
   Rotational movements are limited to prevent excessive tilts:

- **Yaw angle, ψ:**

$$|\psi| \leq 80^\circ \tag{4.24}$$

- **Pitch angle, θ:**

$$|\theta| \leq 40^\circ \tag{4.25}$$

- **Roll angle, ϕ:**

$$|\phi| \leq 40^\circ \tag{4.26}$$

5. **Tether Length and Reel-out Speed**
   The following constraints ensure safe deployment and retrieval of the tether:

   - **Tether length, L_{tether}:**

$$10\,\text{m} \leq L_{\text{tether}} \leq 700\,\text{m} \tag{4.27}$$

   - **Reel-out speed, $\dot{L}_{\text{tether}}$:**

$$-15\,\text{m/s} \leq \dot{L}_{\text{tether}} \leq 20\,\text{m/s} \tag{4.28}$$

6. **Acceleration Constraint**
   The acceleration of the tether release speed, $\ddot{L}_{\text{tether}}$, is restricted to prevent abrupt changes:

$$|\ddot{L}_{\text{tether}}| \leq 2.4\,\text{m/s}^2 \tag{4.29}$$

These constraints are fundamental in ensuring that the kite power system operates within its design parameters, thereby preventing potential failures and optimizing performance throughout the pumping cycle.

4.7 Conclusion

In conclusion, the transition from MATLAB Simulink and standalone CasADi to the integrated AWEBox framework, coupled with the use of IPOPT, provided a more reliable and efficient approach for the simulation and optimization of kite dynamics. The development of custom configuration files within AWEBox allowed us to accurately create a custom model of the specific kite used in the I202 laboratory, illustrated in Figure 1.1, facilitating a wide range of simulation scenarios and optimization tasks. This phase, although time-consuming due to the necessity of ensuring all configurations worked in Python 3.8.19 within a simulated Ubuntu 24.04 LTS environment, laid a strong foundation for further advancements with this powerful tools.

Chapter 5

Simple Closed-Loop Trajectory

In this chapter, we solve the optimal control problem for a simple closed-loop trajectory. The kite starts and finishes within a sphere of 1 unit radius from the initial point after completing one loop. This initial, straightforward simulation was performed to ensure that all simulation parameters are correctly set before moving on to more complex simulations. The primary objective is to verify that the customized kite model functions correctly in solving the optimal control problem.

In this first simulation, there is no reel-out of the tether, the cable remains fixed between the ground and the kite throughout the cycle. The focus is on testing the basic control strategies and kite dynamics. In contrast, the second simulation introduces reel-out, where the kite completes multiple loops up to a predefined altitude.

Additionally, we use a specific type of generator to track and verify the energy produced by the system during the simulation. The cable's diameter, fixed at 2 mm, is chosen to match the characteristics of the model.

5.1 Control and Mathematical Problem for the First Simulation

In the first simulation, we address a simple control problem where the kite follows a closed-loop trajectory with a fixed tether. The objective is to ensure the kite completes one full loop, starting and ending at the same position, without any reel-out of the tether.

5.1.1 Mathematical Problem

The dynamics of the kite are described by a system of differential equations, where the tether length remains constant throughout the flight. This is consistent with the fundamental dynamic equations outlined in previous chapters (see chapter 3 for a detailed formulation). The main variables governing the kite's motion include its elevation angle (β), angular velocity (ω), and the tether tension (T).

These simulations were conducted specifically to analyze the behavior of the kite in a closed-loop trajectory, as introduced earlier. The goal is to explore how the kite's dynamics evolve when completing

a full cycle, maintaining the tether length constant. This approach serves as a preliminary study before introducing reel-out and reel-in dynamics, as discussed in chapter 4.

The ultimate objective, as explained in the previous chapter, is to optimize the kite's trajectory and control strategies to maximize the energy produced during the flight. This closed-loop simulation offers valuable insights into the stability and energy potential of the system, helping refine future optimization strategies to achieve the desired power output.

$$\dot{r} = 0 \quad \text{(tether length is fixed)} \tag{5.1}$$

$$\dot{\beta} = \omega \tag{5.2}$$

$$\dot{\omega} = \frac{1}{mr}\left(-gm\cos(\beta) + F_{\text{aer},\beta}(\alpha)\right) \tag{5.3}$$

$$T = -gm\sin(\beta) + F_{\text{aer},r}(\alpha) + mr\omega^2 \tag{5.4}$$

The challenge lies in stabilizing the kite's trajectory as it moves along the desired path, maintaining the tension and ensuring the kite completes the loop without reel-out.

5.1.2 Control Objective

The primary control objective is to optimize the angle of attack (α) and maintain a stable tether tension (T) throughout the loop. This ensures that the kite follows the desired closed-loop trajectory while adhering to stability and safety constraints. The main control objectives are:

- Ensure the kite completes a closed-loop trajectory starting and ending at the same point with a fixed tether.
- Optimize control inputs (angle of attack α and tether tension T) to maintain flight stability and trajectory accuracy.

5.2 Simulation Setup

The kite operates in a lift mode, generating power during the reel-out phase and consuming energy during the reel-in phase. This cyclical operation is crucial for evaluating the system's efficiency and optimizing the control strategies.

Wind velocity profiles are modeled using a power law to simulate realistic wind conditions at varying altitudes. This method provides a more accurate representation of the wind environment, which is vital for optimizing the kite's trajectory and control strategies. The reference wind speeds and exponents used in the power law are carefully chosen to match real-world conditions as closely as possible.

The optimization problem is formulated to maximize the average power output while ensuring the stability and safety of the system. It is discretized using a zero-order-hold control parameterization,

which simplifies the implementation of control strategies and enhances the numerical stability of the optimization process. IPOPT, an interior point optimizer, is employed as the primary solver to handle nonlinear programming problems (NLP) that arise from the trajectory optimization of the kite. Linear solvers MA57 and MUMPS are used within IPOPT to efficiently manage the Newton steps required for optimization.

Constraints are incorporated into the optimization problem to ensure safe and efficient operation of the kite power system. These constraints include limits on maximum and minimum tether force, airspeed limits, and allowable angles of attack and side-slip. These constraints are essential to maintain the structural integrity of the kite and to ensure stable flight dynamics under varying wind conditions.

5.2.1 Numerical Details

The nonlinear programming problem (NLP) is discretized using a zero-order-hold control parameterization. A simple phase-fixing routine is employed to stabilize the optimization process, with IPOPT used as the primary solver and MA57 as the linear solver for Newton steps. The optimization problem has the following specifics:

Parameter	**Description**
Atmosphere Model	ISA (International Standard Atmosphere)
Wind Model	Power-law
Power-law Exponent	Set to 0.15
Reference Wind Speed	Set to 10 m/s
Reference Altitude	Set to 100 meters
Induction Model	Not in use
System Type	Lift mode
Aircraft DOF	6 degrees of freedom
Number of Aircraft	1
Tether Attachment	Center of mass
Tether Control Variable	Tether reel-out speed ($\dot{l}_t$)
Tether Drag Model	Multi-segment
Tether Drag Elements	5 segments
Ground Station Model	Not included
NLP Solver	IPOPT
Linear Solver	MA57
Collocation Scheme	Radau
Collocation Order	4
Control Parameterization	Zero-order hold (ZOH)
Number of Intervals	500
Phase-fix Strategy	Simple

Table 5.1: Numerical Details of the Optimization Problem used in AWEBox for this simulation.

The model includes constraints such as the maximum and minimum tether force, the maximum and minimum airspeed, and limits on the angles of attack and side-slip.

5.3 Results Analysis

The results obtained from the simulations are analyzed to gain insight into the performance of the system. Key aspects evaluated include the optimized trajectory of the kite, the control actions applied, and the overall power output. The analysis is supported by Figures 5.1 depicts the evolution of state variables, highlighting critical metrics such as tether length and airspeed. Figure 5.2 presents the control inputs applied throughout the cycle. Figure 5.3 shows the generated power, emphasizing the peak outputs and overall efficiency. Figure 5.8 illustrates the optimized flight path of the kite, while Figure 5.4 outlines the constraints applied during simulation, to ensure safe and efficient operation. The comprehensive analysis confirms the system's ability to maintain stability and optimize energy extraction, validating the effectiveness of the implemented optimization strategy.

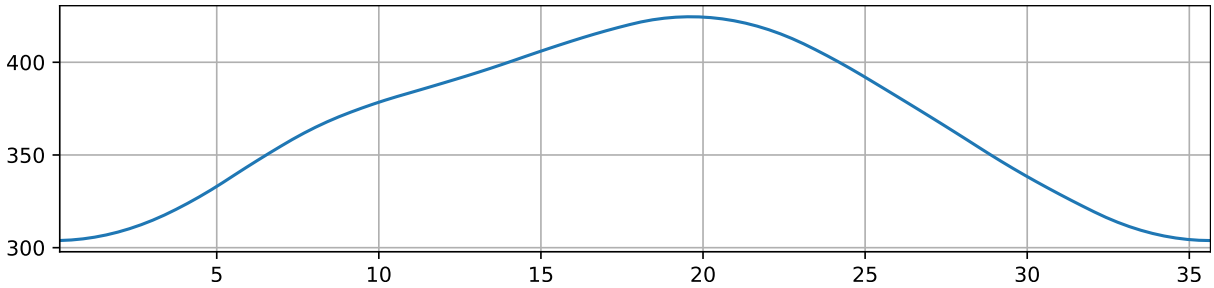

Figure 5.1: Tether Length (r_t): Length of tether deployed, indicating kite altitude.

- **Tether Length** (r_t): This graph displays the length of the tether deployed from the ground station to the kite. It varies during the reel-out and reel-in phases, indicating the kite's altitude changes. The tether length is calculated with respect to the (0, 0, 0) point of the reference coordinate system, ensuring accurate measurement from the ground station to the kite's position.

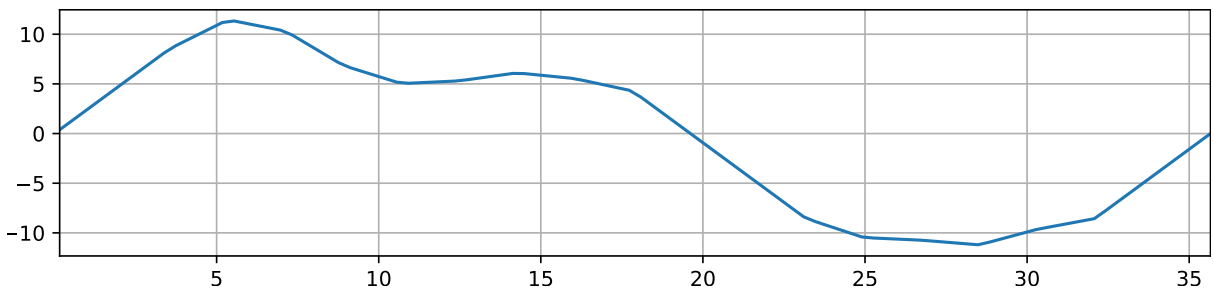

Figure 5.2: Tether Reel-out Speed (v_t): Speed of tether extension/retraction.

- **Tether Reel-out Speed** (v_t): This graph shows the speed at which the tether is being extended or retracted. The variation in speed is critical for optimizing energy generation during the cycle.

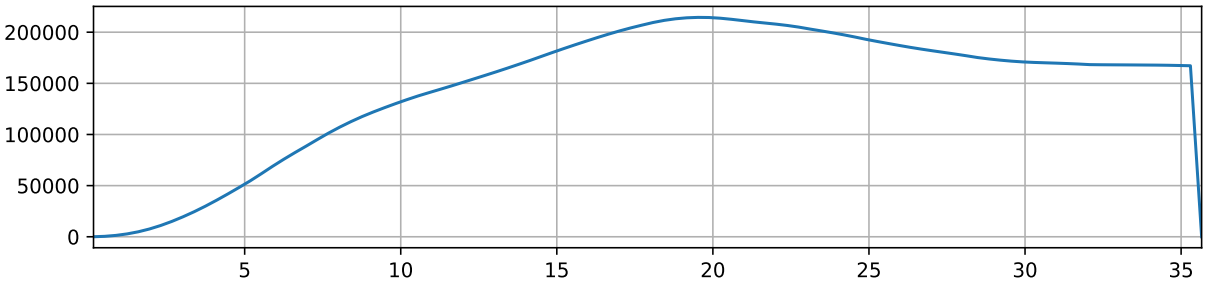

Figure 5.3: Total Energy (e): Total system energy, combining kinetic and potential components.

- **Total Energy** (e): This graph represents the total energy of the system, combining both kinetic and potential energy components. Monitoring this variable is crucial for assessing the efficiency of the energy extraction process.

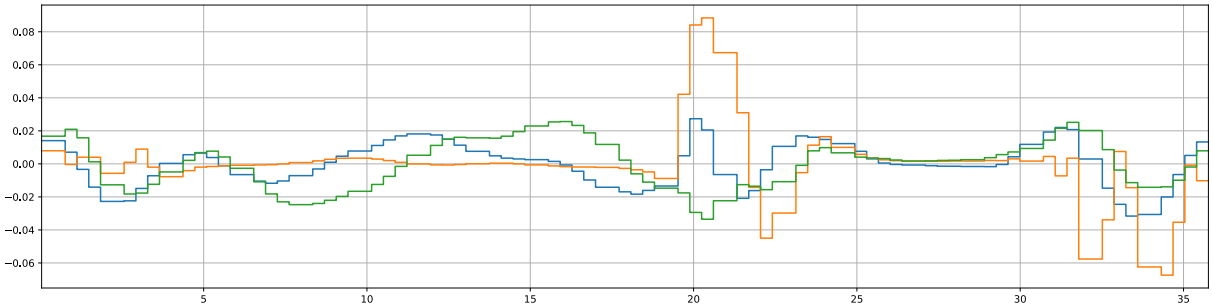

Figure 5.4: Deflection rates ($\dot{\delta}_{\text{aileron}}, \dot{\delta}_{\text{elevator}}, \dot{\delta}_{\text{rudder}}$) during the first simulation. These rates represent the time derivatives of the control surface deflections for the aileron, elevator, and rudder. The yaw angle (ψ, green), pitch angle (θ, blue), and roll angle (ϕ, orange) are plotted to illustrate how the kite's orientation changes throughout the first simulation as a result of these deflections. The goal of this first simulation was to test the stability of the system and ensure that the kite followed a closed-loop trajectory.

The first simulation aimed to assess the stability of the kite's trajectory by evaluating the response of the control surfaces—aileron, elevator, and rudder—through their deflection rates. As shown in Figure 5.4, the deflection rates ($\dot{\delta}_{\text{aileron}}, \dot{\delta}_{\text{elevator}}, \dot{\delta}_{\text{rudder}}$) were crucial for maintaining the desired orientation of the kite throughout the closed-loop trajectory.

In particular, the yaw angle (ψ), pitch angle (θ), and roll angle (ϕ) are plotted to demonstrate the kite's response to control inputs. These angles show how the kite rotates around its respective axes—yaw for directional heading (green), pitch for vertical ascent and descent (blue), and roll for lateral tilt and stability (red).

During the simulation, the primary focus was to ensure that the kite maintained a smooth and stable flight, returning to its initial position after completing the loop. The control system, through the deflection of the aileron, elevator, and rudder, adjusted the kite's aerodynamic properties to ensure that the trajectory remained within acceptable boundaries. Notably, the small variations in the deflection rates reflect fine-tuned control to achieve a balance between aerodynamic forces and the tether tension.

Overall, the first simulation confirmed that the system could follow a closed-loop trajectory with the appropriate control input, without significant deviations or instabilities. This successful outcome laid the foundation for more complex simulations with variable tether lengths and energy optimization in subsequent phases.

5.4 Optimized Trajectory Simulation Results

5.4.1 Overview

This section presents the simulation results for the optimized trajectory of the kite using AWEBox. The optimization aimed to maximize the average power output while ensuring stable and efficient operation. The following figures illustrate the trajectory and key performance metrics.

5.4.2 Simulation Results

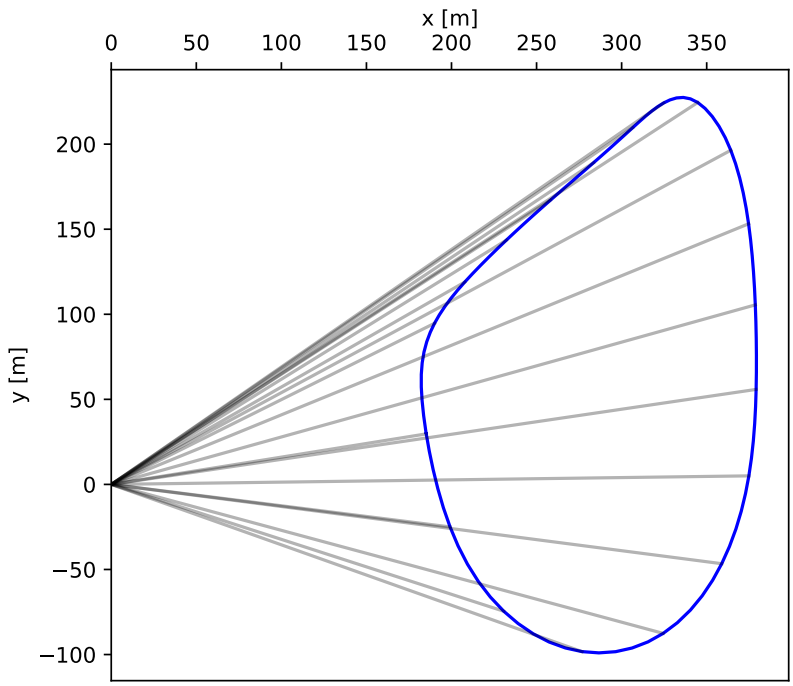

Figure 5.5: Trajectory in the XY plane.

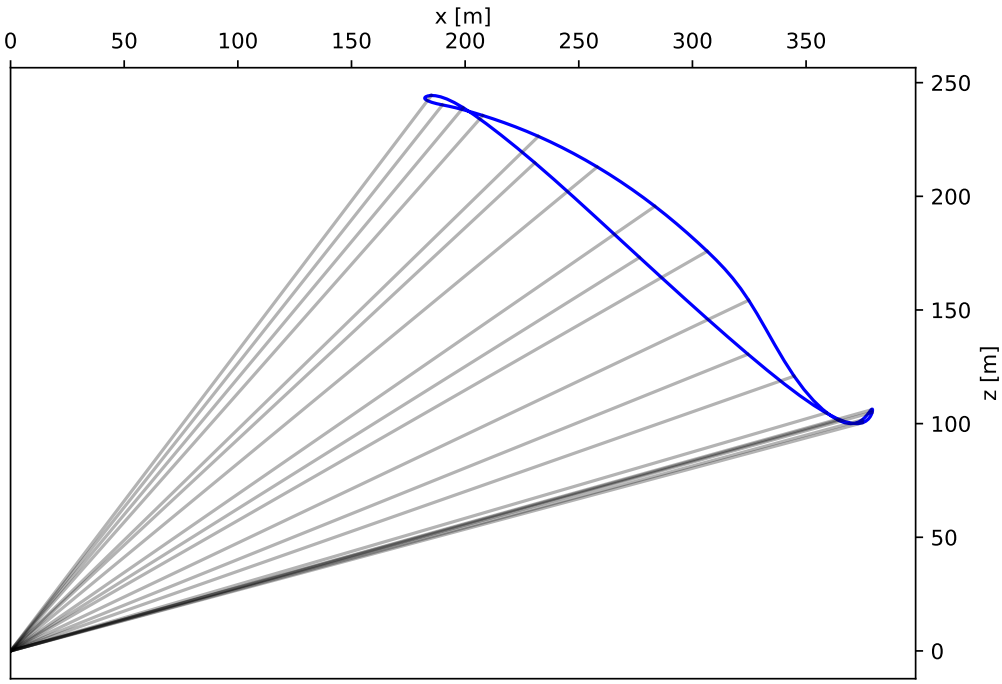

Figure 5.6: Trajectory in the XZ plane.

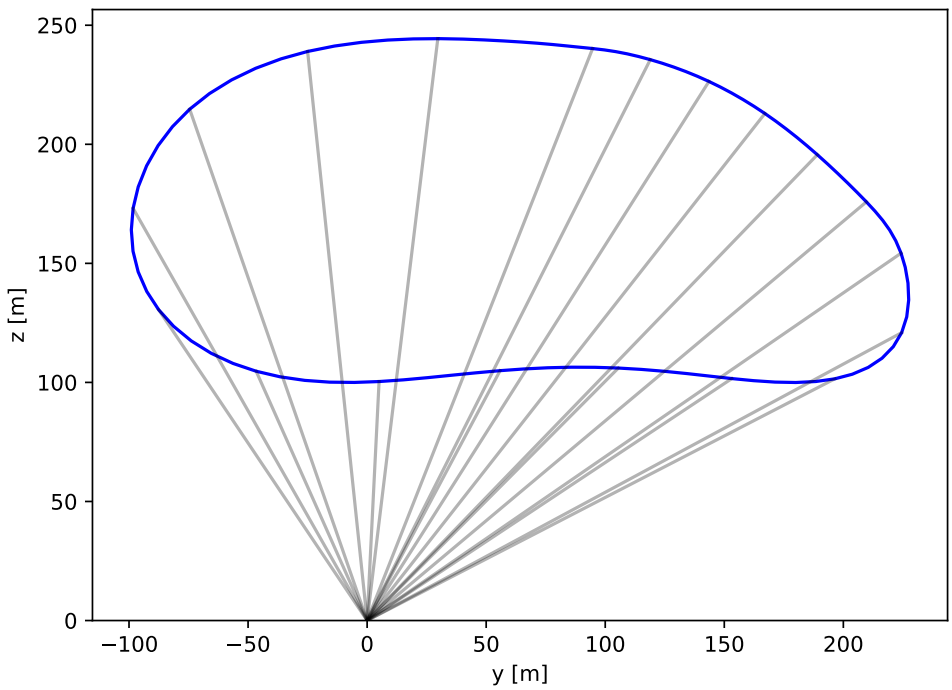

Figure 5.7: Trajectory in the ZY plane.

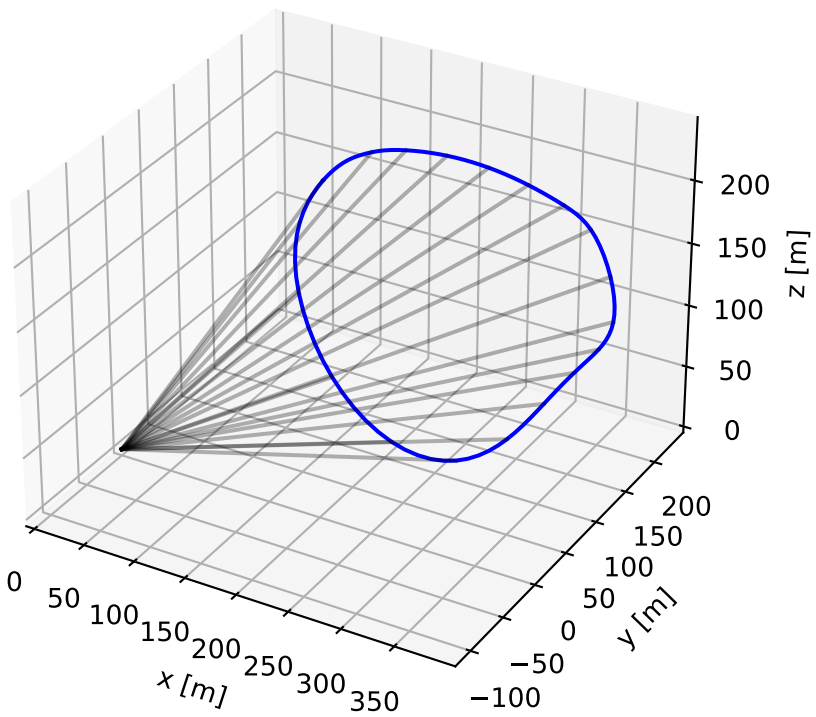

Figure 5.8: Three-dimensional trajectory.

5.4.3 Discussion

The simulation process, using AWEBox, involved modeling the kite's aerodynamic and tether dynamics, as well as defining control inputs and strategies. The key settings used include a 6-DOF (degrees-of-freedom) model for the aircraft, the use of the "power" wind model, and the "direct-collocation" method for optimization. The optimization process was completed in approximately 19.4 seconds, achieving an average power output of 4.6856 kW over a time period of 35.6575 seconds.

The use of AWEBox allowed for precise modeling of the kite's dynamics and the optimization of the control strategies. This was critical in ensuring the stability and efficiency of the operation, with all tests confirming the numerical soundness of the solution.

5.4.4 Conclusion

The optimized trajectory simulation of the kite using AWEBox successfully demonstrated the system's capability to maintain a stable and efficient operation. The simulation confirmed that the initial and final tether lengths were almost equal, indicating a complete and balanced cycle. The kite maintained an optimal trajectory throughout the simulation, maximizing the average power output to 4.6856 kW over a period of 35.6575 seconds. This result underscores the effectiveness of the implemented optimization strategies and the accurate modeling of aerodynamic and tether dynamics within the AWEBox framework. The consistency in the initial and final tether lengths also suggests that the control strategies were effective in managing the energy extraction process, ensuring that the system could reliably cycle

without excessive wear or energy loss. These findings validate the potential of using advanced control strategies in airborne wind energy systems to optimize power generation and system stability.

In the next chapter, we will simulate advanced trajectories, exploring more complex maneuvers and optimizing further aspects of the system's performance. These simulations aim to push the boundaries of current control strategies and assess their effectiveness in more challenging operational scenarios.

Chapter 6

Closed-Loop Trajectory with Ascending and Retracting Phase

6.1 Introduction

This chapter addresses the simulation of advanced closed-loop trajectories for kite power systems, focusing on the ascending (tether reel-out) and retracting (tether reel-in) phases. The problem is formulated using the AWEBox framework, which provides a realistic and highly detailed solution to the reel-out and reel-in processes, simulating them in a visually and dynamically accurate manner.

The optimal control problem is twofold: during the ascent, the objective is to maximize power output by adjusting the control inputs such as reel-out speed and angle of attack. During the retraction phase, the goal is to minimize the time taken to reel in the tether while maintaining stability and minimizing energy losses.

The full cycle consists of three phases:

- **Traction Mode:** The kite ascends, and the tether is reeled out, extracting energy from the wind.
- **Transition Mode:** A brief phase where the kite transitions between ascent and descent.
- **Retraction Mode:** The tether is reeled in, and the goal is to minimize the time and energy spent in this phase.

Additionally, the tether is divided into 5 segments, each contributing to the system's overall dynamics and performance. This segmentation allows for a more accurate representation of tether drag and other forces acting on the kite throughout the cycle.

The simulation builds on previous methodologies discussed in chapter 3, introducing more complex dynamics and control strategies tailored to the unique challenges of each phase of the cycle. The results of this simulation aim to optimize the energy cycle by refining the control strategies across all three phases.

6.2 Control and Mathematical Problem for the Second Simulation

In this simulation, we address the optimal control problem for a kite power system following a closed-loop trajectory, with the tether length dynamically changing across three distinct phases: traction (reel-out), transition, and retraction (reel-in). The objective is to manage the kite's trajectory and tether dynamics to maximize power output during the ascent, ensure stability during the transition, and minimize retraction time while maintaining system stability.

6.2.1 Traction Mode (Reel-Out Phase)

During this phase, the tether length increases as the kite ascends. The primary control objective is to maximize the power extracted from the wind by optimizing the reel-out speed v_t and the angle of attack α. The dynamics governing this phase are described in subsection 4.6.1.

Objective: Minimize the cost functional $J(x,u)$:

$$\text{Minimize} \quad J(x,u) = -\int_{t_0}^{t_f} \dot{r}T \, \mathrm{d}t \tag{6.1}$$

The total energy produced over the time period is:

$$E(t_f) = \int_0^{t_f} P(t) \, \mathrm{d}t \tag{6.2}$$

where the instant power production $P(t)$ is given by:

$$P(t) = -v_t T \tag{6.3}$$

In order to focus the energy production at the beginning of the time interval, we modify the objective function to:

$$\text{Minimize} \quad -\int_0^{10} e^{-t} \dot{r}T \, \mathrm{d}t \tag{6.4}$$

- **Tether Length and Reel-out Speed**
  The following constraints ensure safe deployment and retrieval of the tether:
  - **Tether length, L_{tether}:**
  $$200\,\text{m} \leq L_{\text{tether}} \leq 400\,\text{m} \tag{6.5}$$
  - **Reel-out speed, $\dot{L}_{\text{tether}}$:**
  $$-15\,\text{m/s} \leq \dot{L}_{\text{tether}} \leq 20\,\text{m/s} \tag{6.6}$$

- **Acceleration Constraint**
  The acceleration of the tether release speed, $\ddot{L}_{\text{tether}}$, is restricted to prevent abrupt changes:

$$|\ddot{L}_{\text{tether}}| \leq 2.4\,\text{m/s}^2 \tag{6.7}$$

The power output P_{out} is maximized during this phase by adjusting v_t and α, subject to aerodynamic constraints.

6.2.2 Transition Mode

This is a brief phase where the kite transitions from the ascending (traction) phase to the descending (retraction) phase. The tether length is momentarily held constant, and the focus is on ensuring a smooth transition by adjusting the control inputs to stabilize the system dynamics.

6.2.3 Retraction Mode (Reel-In Phase)

During the retraction phase, the tether is reeled in, and the objective is to minimize the time required to complete this phase while maintaining stability and minimizing energy consumption. The dynamics are similar to the reel-out phase, but the tether length decreases:

$$\dot{r} = -v_t \quad \text{(tether reel-in speed, control input)} \tag{6.8}$$

$$\dot{\beta} = \omega \tag{6.9}$$

$$\dot{\omega} = \frac{1}{mr}\left(-gm\cos(\beta) + F_{\text{aer},\beta}(\alpha) + 2mv_t\omega\right) \tag{6.10}$$

$$E_{\text{reel-in}} = \int_0^{t_{\text{reel-in}}} T v_t \, dt \tag{6.11}$$

The control inputs v_t and α are adjusted to minimize the retraction time $t_{\text{reel-in}}$ and reduce energy consumption during this phase. In this simulation, a key constraint was imposed to ensure that the final state of the kite remains within a specified tolerance around the target final state. Mathematically, this is expressed as:

$$(r_{\text{end}} - r_f)^2 \leq \text{tolerance_radius}^2 \tag{6.12}$$

where r_{end} represents the final position vector of the kite, r_f is the desired final position, and tolerance_radius denotes the allowable radius around the target final position. For this simulation, the tolerance radius was set to 1 unit. This constraint ensures that the kite returns close to its starting position within a defined spherical region, thereby maintaining a closed-loop system. The implementation of this constraint was crucial for achieving accurate and reliable control in the simulation, ensuring the kite's trajectory remained within acceptable bounds.

6.2.4 Numerical Details

The numerical setup for the non-linear programming problem (NLP) involves several key configurations to ensure accurate and efficient optimization. The NLP is discretized using a zero-order-hold (ZOH) control parameterization, which helps to maintain consistency in control inputs over each interval. A simple phase-fixing routine is utilized to stabilize the optimization process, making it more robust against convergence issues. IPOPT serves as the primary solver for the NLP, with MA57 employed as the linear solver for Newton steps. The optimization problem has the following specifics:

Parameter	**Description**
Atmosphere Model	ISA (International Standard Atmosphere)
Wind Model	Power-law
Power-law Exponent	Set to 0.15
Reference Wind Speed	Set to 10 m/s
Reference Altitude	Set to 100 meters
Induction Model	Not in use
System Type	Lift mode
Aircraft DOF	6 degrees of freedom
Number of Aircraft	1
Tether Attachment	Center of mass
Tether Control Variable	Tether reel-out speed ($\dot{l}_t$)
Tether Drag Model	Multi-segment
Tether Drag Elements	5 segments
Ground Station Model	Not included
NLP Solver	IPOPT
Linear Solver	MA57
Collocation Scheme	Radau
Collocation Order	4
Control Parameterization	Zero-order hold (ZOH)
Number of Intervals	500
Phase-fix Strategy	Simple
Initial Tether Length	50 m
Final Tether Length	50 m
Number of Windings	4

Table 6.1: Numerical Details of the Optimization Problem used in AWEBox for this simulation.

The model incorporates several constraints, such as the maximum and minimum tether force, air-speed limits, and limits on angle of attack and side-slip, ensuring safe and efficient operation. The initial conditions are carefully set, with an initial tether length of 50 meters and a final length of 50 meters, allowing for a comprehensive examination of the kite's dynamic behavior throughout the ascent and descent phases.

The script's implementation includes the specification of the environment, particularly the wind profile, characterized by a power-law model with a reference wind speed of 10 m/s at 100 meters altitude.

The setup also involves adjusting the initial conditions for the kite's state variables, such as groundspeed and inclination, to ensure realistic simulation conditions.

The optimization process iteratively adjusts the system's variables to maximize the average power output while adhering to all imposed constraints. The results, including trajectory plots and power output over time, provide insight into the system's performance under the given conditions. The comprehensive analysis ensures the numerical soundness of the solution, validated by passing all quality checks. The average power output achieved is approximately 4.971 kW, with a pumping cycle duration of 60 seconds and a tether diameter optimized to 2 mm.

6.3 Implementation and Results

The optimization problem was constructed in AWEBox, and the simulations were executed to analyze the performance of the kite. Key performance indicators, such as average power output, trajectory stability, and control variable behavior, were evaluated.

The control variables, including tether length (Figure 6.1), reel-out speed (Figure 6.2), and total energy (Figure 6.3), are essential for optimizing the kite's trajectory and maintaining stability throughout the cycle.

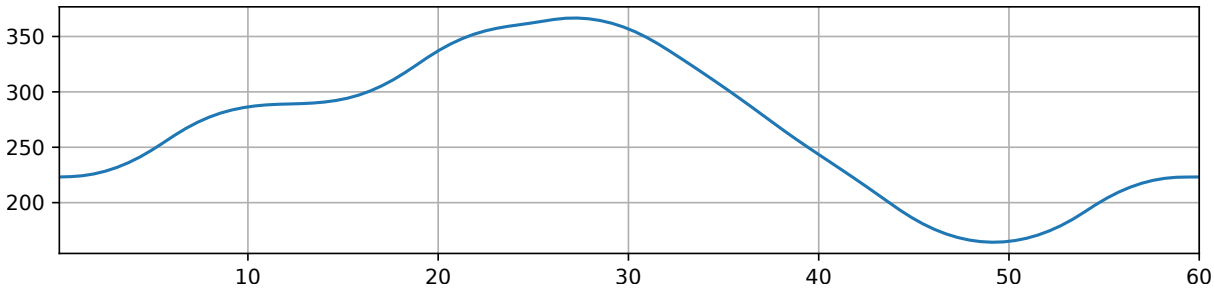

Figure 6.1: Tether Length (r_t): Length of tether deployed, indicating kite altitude.

- **Tether Length** (r_t): This graph displays the length of the tether deployed from the ground station to the kite. It varies during the reel-out and reel-in phases, indicating the kite's altitude changes. The tether length is calculated with respect to the (0, 0, 0) point of the reference coordinate system, ensuring accurate measurement from the ground station to the kite's position.

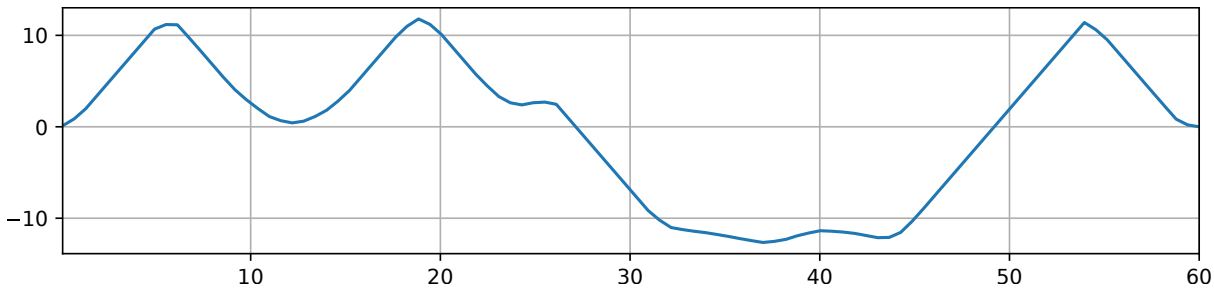

Figure 6.2: Tether Reel-out Speed (v_t): Speed of tether extension/retraction.

- **Tether Reel-out Speed** (v_t): This graph shows the speed at which the tether is being extended or retracted. The variation in speed is critical for optimizing energy generation during the cycle.

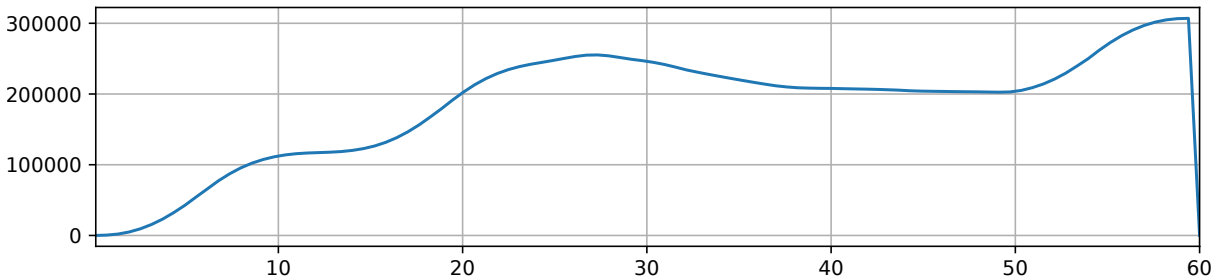

Figure 6.3: Total Energy (e): Total system energy, combining kinetic and potential components.

- **Total Energy** (e): This graph represents the total energy of the system, combining both kinetic and potential energy components. Monitoring this variable is crucial for assessing the efficiency of the energy extraction process.

6.3.1 Control Variables Over Time

The control variables, including tether length (Figure 6.1), reel-out speed (Figure 6.2), and total energy (Figure 6.3), are essential for optimizing the kite's trajectory and maintaining stability throughout the cycle.

This figure illustrates the control inputs applied to the system, including the reel-out speed of the tether and the angular rates. These control variables are adjusted to optimize the kite's trajectory and maintain stability throughout the pumping cycle. The zero-order hold (ZOH) control parameterization ensures discrete control actions at each interval, contributing to the overall efficiency of the system.

6.3.2 Power Output Over a Complete Cycle

Figure 6.4 demonstrate the system's control inputs, including the deflection rate and tether reel-out acceleration, while also illustrating the power output generated during the cycle.

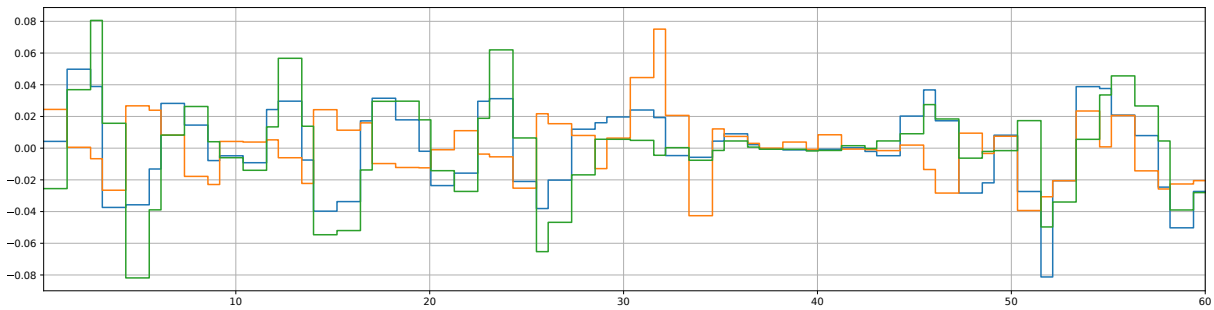

Figure 6.4: Deflection rates ($\dot{\delta}_{\text{aileron}}, \dot{\delta}_{\text{elevator}}, \dot{\delta}_{\text{rudder}}$) for the control surfaces of the kite, representing the rate of change of the aileron, elevator, and rudder deflections. The yaw angle (ψ, green), pitch angle (θ, blue), and roll angle (ϕ, orange) are shown in the graph, illustrating how the kite's orientation changes over time as a result of the deflections.

In the figure, the control surfaces for the kite are depicted by the following deflections:

$$\delta_k = [\delta_{\text{aileron}}, \delta_{\text{elevator}}, \delta_{\text{rudder}}] \in R^3 \tag{6.13}$$

These deflections correspond to the aileron, elevator, and rudder, respectively, and are used to control the aerodynamic behavior of the kite. The yaw angle (ψ), pitch angle (θ), and roll angle (ϕ) shown in the figure are key indicators of the kite's orientation during flight:

- **Yaw angle**, ψ (green in the figure): indicates the rotation of the kite around its vertical axis, affecting its directional heading.
- **Pitch angle**, θ (blue in the figure): represents the rotation around the lateral axis, controlling ascent and descent.
- **Roll angle**, ϕ (orange in the figure): describes the rotation around the longitudinal axis, which influences the kite's banking and stability.

The state vectors used in the simulation, which are crucial for controlling these angles, are defined as:

$$\mathbf{x}_{\text{lift}} = (\mathbf{x}_{\text{base}}, l_t, \dot{l}_t) \tag{6.14}$$

and

$$\mathbf{x}_{\text{drag}} = (\mathbf{x}_{\text{base}}, \kappa) \tag{6.15}$$

Where:

- l_t is the tether length and $\dot{l}_t$ is the reel-out/reel-in speed, both of which are represented in the figure.
- κ is the aerodynamic drag coefficient, essential for controlling the drag-mode dynamics.

6.3.3 Trajectory Plot of the Kite

Figures 6.5 to 6.9 present the optimized 3D trajectory of the kite, including projections in the XY, XZ, and ZY planes, as well as a detailed three-dimensional view. These plots provide a comprehensive visualization of the kite's path, showcasing the effectiveness of the control strategies in maximizing wind energy capture while maintaining operational safety limits.

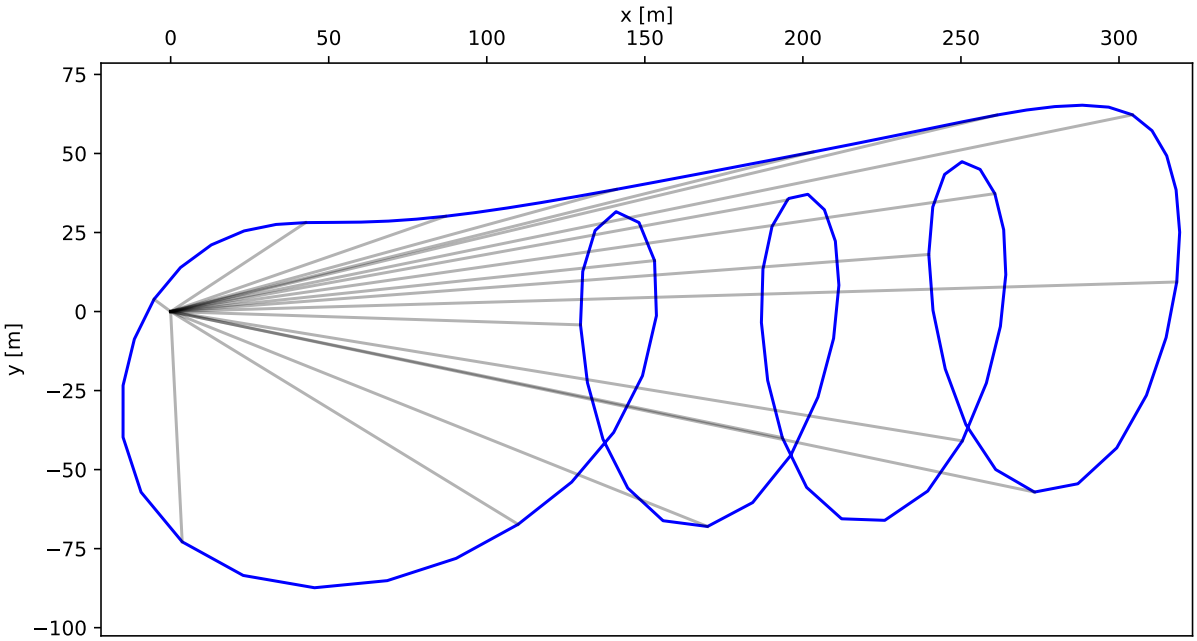

Figure 6.5: Trajectory in the XY plane.

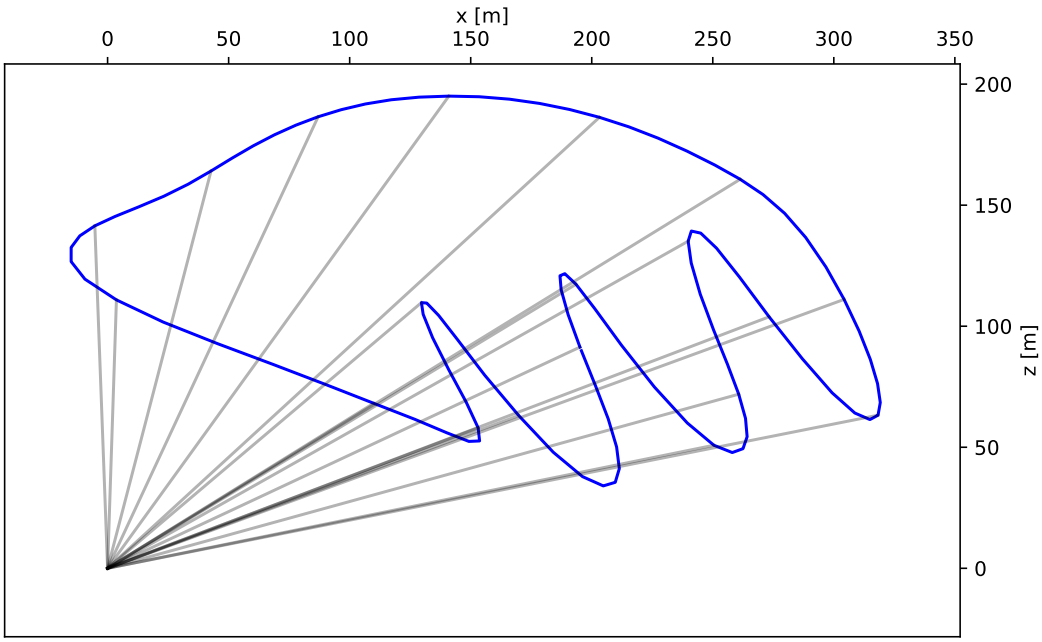

Figure 6.6: Trajectory in the XZ plane.

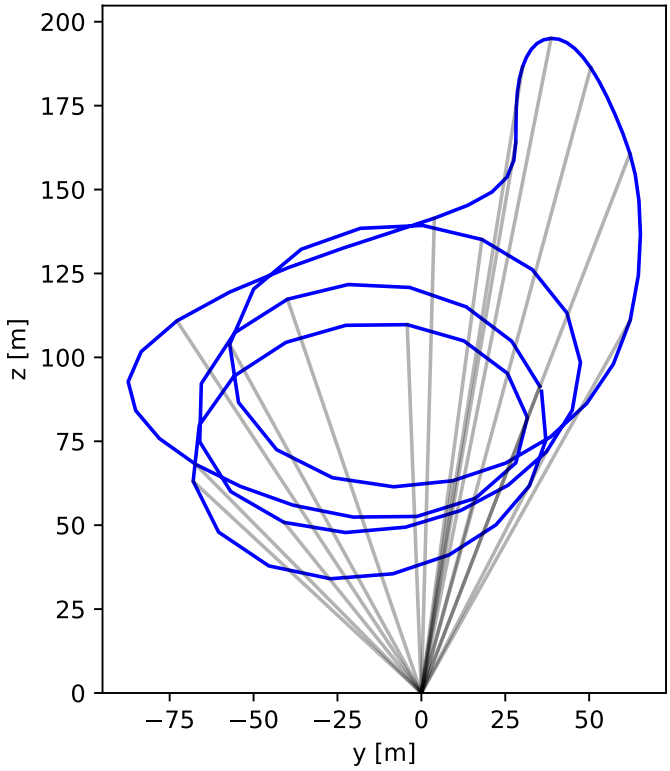

Figure 6.7: Trajectory in the ZY plane.

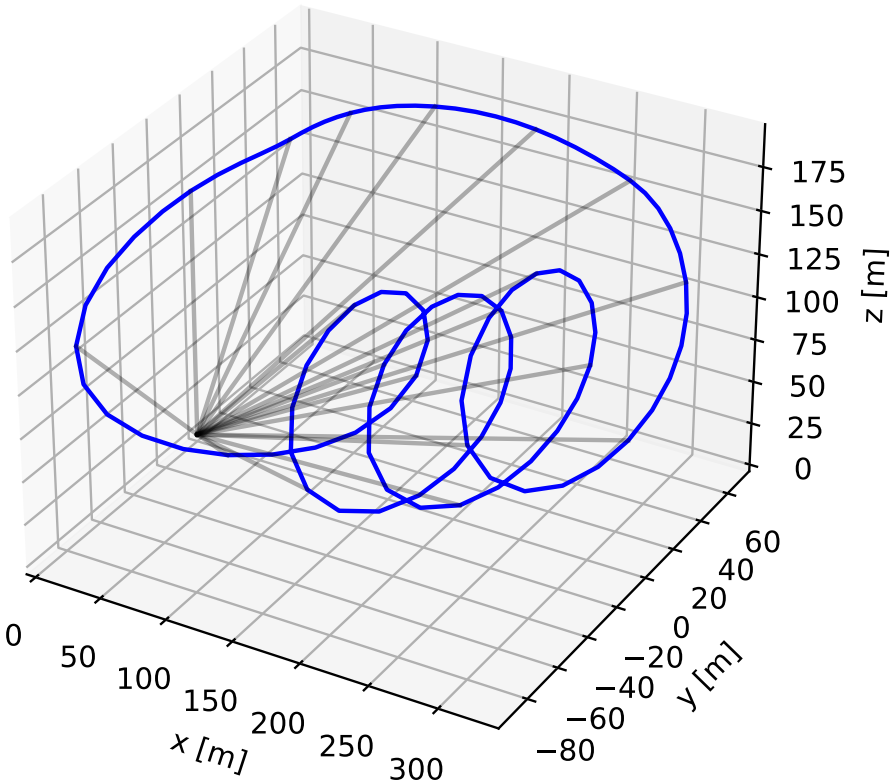

Figure 6.8: Three-dimensional trajectory.

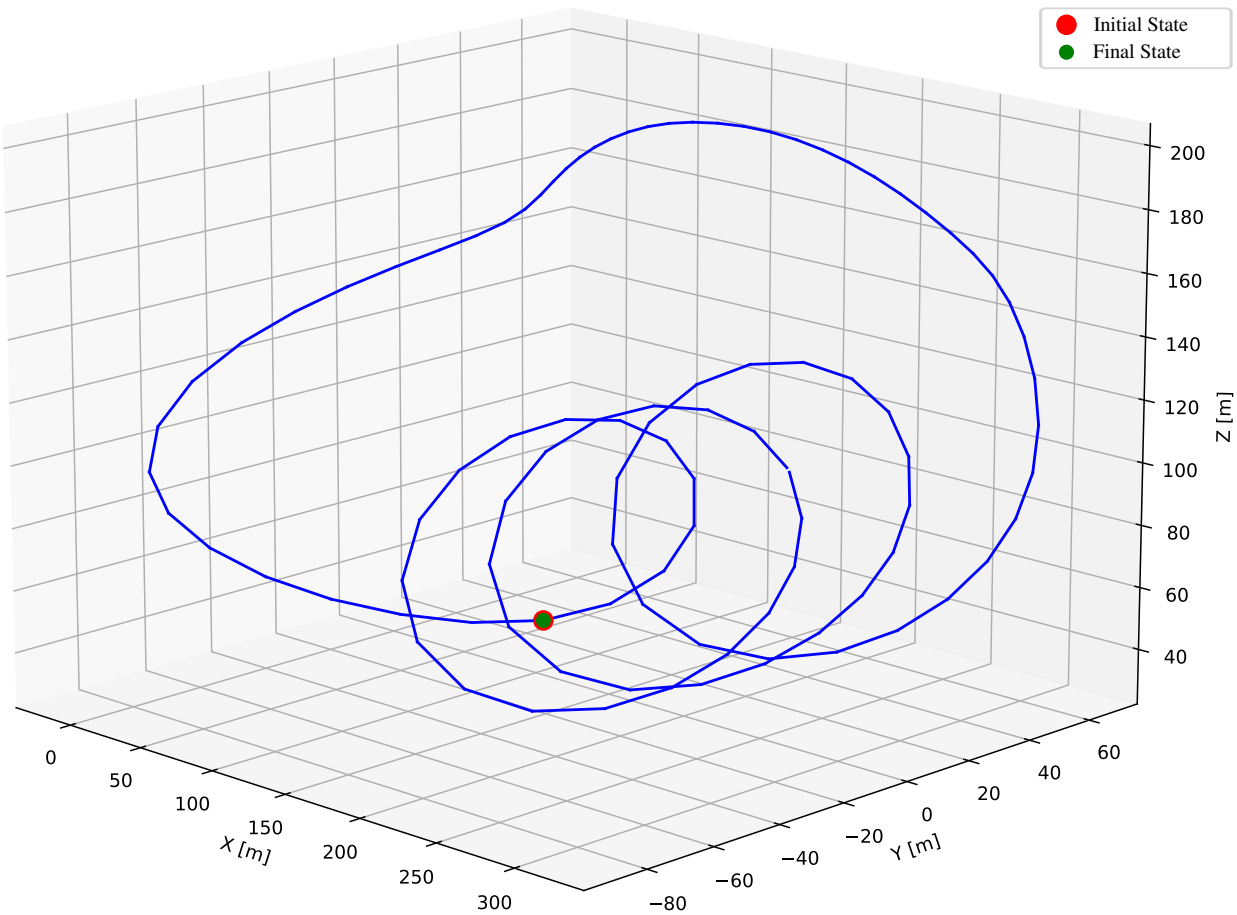

Figure 6.9: Detailed three-dimensional trajectory. The trajectory starts at the *initial state* (green) and ends at the *final state* (red), following an upward path and then returning to the initial point.

This figure shows the 3D trajectory of the kite during the pumping cycle. The kite follows a carefully designed circular path in the sky, in order to optimize wind energy capture. The trajectory is meticulously planned to maximize wind exposure while ensuring that operational constraints, such as tether length and angular limits, are strictly adhered to.

In this simulation, an optimal trajectory was established with a key constraint: the final position of the kite must lie within a sphere of 1 unit radius from the initial point. This constraint ensures that the kite returns close to its starting position, maintaining a closed-loop system. The trajectory involves four complete windings, starting from an initial elevation of 50 units above the ground. The kite ascends to a maximum altitude, utilizing wind energy, before retracting back to the final position. This ascending and retracting motion allows efficient energy generation during ascent and minimal energy loss during descent, showcasing the effectiveness of the designed control strategy.

The 3D trajectory clearly depicts the circular path as the kite ascends, followed by a distinct transition at the top point, where the kite begins its descent back to the initial position. This well-defined circular motion ensures optimal utilization of the available wind energy, demonstrating the precision of

the control strategy and the kite's ability to adhere to the planned path.

6.3.4 Discussion

The simulation process, using AWEBox, focused on optimizing the kite for a single-kite airborne wind energy system. The model featured a 6-DOF aircraft configuration with key constraints, such as tether force limits and airspeed limits. The optimization utilized the "direct-collocation" method with Radau collocation points and a zero-order hold (ZOH) control parameterization. The IPOPT solver, combined with the MA57 linear solver, efficiently handled the nonlinear programming (NLP) problem.

The process included multiple phases, such as minimizing fictitious forces and maximizing average power output. The simulation achieved an average power output of 4.97138 kW over a 60-second cycle, demonstrating effective control and energy management. The entire computation process took approximately 1 hour and 21 minutes, indicating a significant computational effort to achieve the optimized solution.

6.3.5 Conclusion

The optimized simulation demonstrated the kite capability to maintain stable and efficient operation. The results indicated that the kite followed a well-defined trajectory, adhered to operational constraints, and maximizing power output. The consistent numerical soundness, as confirmed by all quality checks, underscores the effectiveness of the strategies implemented. The simulation validated the potential of advanced optimization techniques in enhancing the performance and reliability of airborne wind energy systems.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Throughout this dissertation, I have explored the complexities and challenges of optimizing and simulating Airborne Wind Energy Systems (AWES). The journey provided valuable insights and learning experiences, which are summarized as follows.

- **Deep Understanding of AWES Dynamics:** Working on this project allowed me to gain a comprehensive understanding of the dynamics involved in kite-based energy systems. This included learning about the aerodynamics, control systems, and the critical factors that influence the performance of AWES.
- **Hands-on Experience with Simulation Tools:** I acquired significant experience with simulation tools and frameworks, such as awebox and CasADi. This hands-on experience was invaluable in understanding how to model complex systems and optimize their performance. The challenges faced with the integration of CasADi and Python also highlighted the importance of choosing the right tools for specific tasks.
- **Problem-Solving Skills:** The process of developing and optimizing the closed-loop trajectory required strong problem-solving skills. I learned to identify and address issues related to system stability, energy capture, and control strategies. This process also taught me the importance of iterative testing and refinement in achieving optimal results.
- **Adaptability in Research:** The difficulties encountered with certain tools and methodologies taught me to be adaptable and open to exploring alternative solutions. This flexibility was crucial in overcoming obstacles and ensuring the success of the project.
- **Critical Analysis and Evaluation:** An essential part of the project was critically analyzing the results and evaluating the system's performance under various conditions. This aspect of the

research helped me develop a keen eye for detail and a better understanding of the importance of thorough evaluation in engineering projects.

The experiences and lessons learned during this dissertation not only have deepened my technical knowledge but also improved my research and analytical skills, preparing me for future challenges in the field of renewable energy systems. Throughout the process, I have gained a comprehensive understanding of the complexities involved in designing, simulating, and optimizing airborne wind energy systems (AWES). This journey has provided me with valuable insight into advanced control systems, computational modeling, and the integration of multidisciplinary approaches to solve complex engineering problems.

In addition to the technical aspects, I have also developed critical soft skills such as project management, teamwork, and effective communication. These skills have proven essential for navigating the challenges of a large-scale research project, from initial conception to final implementation. The ability to communicate complex ideas clearly and concisely has been particularly important in collaborating with peers, advisors, and industry professionals.

The knowledge and skills gained will be invaluable for my future endeavors in the field of renewable energy. As the world continues to search for sustainable energy solutions, I am eager to apply what I have learned to real-world applications, contribute to innovative projects, and push the boundaries of what is possible in renewable energy technologies. This dissertation has been a pivotal step in my academic and professional journey, laying a solid foundation for future research and development in the ever-evolving landscape of sustainable energy.

7.2 Future Work

Reflecting on the work accomplished, there are several directions for future research that could be built on the findings of this dissertation.

- **Enhanced Control Systems:** Future work could focus on developing more sophisticated control algorithms that can adapt to changing environmental conditions in real-time. This would further optimize the efficiency and stability of AWES.
- **Full Autonomy:** Achieving full autonomy for AWES, including automatic takeoff, landing, and energy production cycles, is a significant area for future exploration. This would reduce the need for human intervention and potentially increase the system's efficiency and reliability.
- **Material Innovations:** Research into advanced materials with higher strength-to-weight ratios and improved durability could enhance the performance and lifespan of AWES components.
- **Scalability and Integration Challenges:** Investigating the challenges associated with scaling AWES to larger systems and integrating them into existing energy grids will be crucial for widespread adoption.
- **In-depth Performance Evaluation:** Continued efforts in evaluating the environmental and economic impacts of AWES, including comprehensive life cycle assessments, will be important for understanding their long-term viability and benefits.

In conclusion, the dissertation journey has been both challenging and rewarding. The knowledge and skills gained will be invaluable for my future endeavors in the field of renewable energy. There is still much to explore and improve in the domain of AWES, and I am eager to continue to contribute to this evolving field. The experiences of this project not only have solidified my understanding of AWES but also have fueled my passion for innovation and research within the renewable energy sector. As I move forward, I am excited to explore new frontiers and challenges, leveraging the insights and expertise developed during this dissertation.

Bibliography

[1] UPWIND Research Project. *UPWIND Research Project*. https://www.upwindresearchproject.com/. 2024.

[2] IPOPT Development Team. *IPOPT: Interior Point Optimizer*. https://coin-or.github.io/Ipopt/.

[3] Coin-HSL. *Coin-HSL: A Collection of Fortran Codes for Large-Scale Scientific Computation*. https://licences.stfc.ac.uk/product/coin-hsl.

[4] Joel A. E. Andersson et al. "CasADi – A Software Framework for Nonlinear Optimization and Optimal Control". In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36. DOI: 10.1007/s12532-018-0139-4. URL: https://doi.org/10.1007/s12532-018-0139-4.

[5] J. De Schutter et al. "AWEbox: An Optimal Control Framework for Single- and Multi-Aircraft Airborne Wind Energy Systems". In: *Energies* 16.4 (2023), p. 1900. DOI: 10.3390/en16041900. URL: https://doi.org/10.3390/en16041900.

[6] European Environment Agency. *EU Achieves 20-20-20 Climate and Energy Targets*. https://www.eea.europa.eu/highlights/eu-achieves-20-20-20. 2020.

[7] Global Wind Energy Council. *Global Wind Energy Council*. https://gwec.net/.

[8] David de Meij. *Why the Future of Wind Energy is Airborne*. https://medium.com/@daviddemeij/why-the-future-of-wind-energy-is-airborne-f77d08abbeed. 2023.

[9] Airborne Wind Europe. *Benefits of Airborne Wind Energy*. https://airbornewindeurope.org/benefits/.

[10] Lorenzo Fagiano, Marco Milanese, and Dario Piga. "Optimization of Airborne Wind Energy Generators for a Pumping Cycle Generation". In: *Wind Energy Science Discussions* (2020), pp. 1–18. URL: https://wes.copernicus.org/preprints/wes-2020-123/wes-2020-123.pdf.

[11] CT Engineering Group. *7 Reasons to Use AWES Technology*. https://www.ctengineeringgroup.com/7-reasons-to-use-awes-technology/.

[12] ResearchGate. *Fly-Gen AWESs: Summary of FG-AWESs*. https://www.researchgate.net/figure/Fly-Gen-AWESs-Summary-of-FG-AWESs_fig13_282314959.

[13] ResearchGate. *Different Types of Aircraft in Fly-Gen Systems*. https://www.researchgate.net/figure/Different-types-of-aircraft-in-Fly-Gen-systems-a-Plane-with-four\-turbines-design-by_fig8_282314959.

[14] Alberto Cherubini et al. "Airborne Wind Energy Systems: A Review of the Technologies". In: *Energy Conversion and Management* 227 (2021), p. 113648. DOI: 10.1016/j.enconman.2020.113648. URL: https://www.sciencedirect.com/science/article/abs/pii/S019689042100090X.

[15] Author(s) Name. *Title of the Paper*. http://portalabepro.educacao.ws/ijcieom/restrito/arquivos/icieom2020/FULL_0010_37171.pdf. International Joint Conference on Industrial Engineering and Operations Management (ICIEOM). 2020.

[16] A. Cherubini et al. "Airborne Wind Energy Systems: A review of the technologies". In: *Renewable and Sustainable Energy Reviews* 51 (2015), pp. 1461–1476.

[17] European Sustainable Energy Week. *Airborne Wind Energy Project Announced as Finalist for the 2024 European Sustainable Energy Awards*. https://sustainable-energy-week.ec.europa.eu/news/airborne-wind-energy-project-announced-finalist-2024-european\-sustainable-energy-awards-2024-05-07_en. 2024.

[18] AWESCO Consortium. *AWE Explained*. https://awesco.eu/awe-explained/.

[19] AWESCO. *AWE Explained*. https://awesco.eu/awe-explained/. 2024.

[20] Michael Erhard and Stephan Horn. "Optimal Control Algorithms with Adaptive Time-Mesh Refinement for Kite Power Systems". In: *Renewable Energy* 114 (2018), pp. 1124–1133. DOI: 10.1016/j.renene.2017.07.095. URL: https://www.researchgate.net/publication/323418452_Optimal_Control_Algorithms_with_Adaptive_Time-Mesh_Refinement_for_Kite_Power_Systems.

[21] Joaquim José da Silva. "Development and Control of an Airborne Wind Energy Pumping System". Master's Thesis. University of Porto, 2018. URL: https://repositorio-aberto.up.pt/bitstream/10216/114178/2/278042.pdf.

[22] Lawrence C. Evans. *An Introduction to Mathematical Optimal Control Theory*. https://math.berkeley.edu/~evans/control.course.pdf. 2010.

[23] Bernd Flemisch et al. "A Coupling Approach for Multiphase Flow in Fractured Porous Media". In: *Modeling, Simulation and Optimization of Complex Processes*. Ed. by Andreas Griewank et al. De Gruyter, 2011, pp. 157–166. DOI: 10.1515/9783110249996.157. URL: https://www.degruyter.com/document/doi/10.1515/9783110249996.157/html.

[24] Artelys. *Knitro Solver*. https://www.artelys.com/solvers/knitro/.

[25] WORHP. *WORHP: We Optimize Really Huge Problems*. https://worhp.de/.

[26] Andreas Wächter, Jeffrey T. Linderoth, and Sarah M. Shoemaker. "Exploring Benefits of Linear Solver Parallelism on Modern Nonlinear Optimization Applications". In: *Optimization and Engineering* 21 (2020), pp. 1011–1033. DOI: 10.1007/s11081-019-09470-4. URL: https://www.researchgate.net%20publication/335908712_Exploring_Benefits_of_Linear_Solver_Parallelism_on_Modern_Nonlinear_Optimization_Applications.

[27] Imperial College London. *ICLOCS: Imperial College London Optimal Control Software*. http://www.ee.ic.ac.uk/ICLOCS/.

[28] AMPL Optimization Inc. *AMPL: A Mathematical Programming Language*. https://ampl.com/.

[29] HSL. *MA57 - A Code for the Solution of Sparse Symmetric Definite and Indefinite Systems*. https://www.hsl.rl.ac.uk/catalogue/ma57.html.

Appendix A

Appendix A

Auxiliary Equations

The equations for the 2D and 3D models are derived in this chapter.

A.1 2D Model

The position vector **p** in a 2D plane is given by:

$$\mathbf{p} = \begin{bmatrix} r\cos(\beta) \\ r\sin(\beta) \end{bmatrix} \tag{A.1}$$

The time derivative of the position vector, representing velocity, is:

$$\begin{aligned}
\dot{\mathbf{p}} &= \frac{d\mathbf{p}}{dt} && \text{(A.2)}\\
&= \frac{\partial \mathbf{p}}{\partial r}\frac{dr}{dt} + \frac{\partial \mathbf{p}}{\partial \beta}\frac{d\beta}{dt} && \text{(A.3)}\\
&= \frac{\partial \mathbf{p}}{\partial r}\dot{r} + \frac{\partial \mathbf{p}}{\partial \beta}\dot{\beta} && \text{(A.4)}\\
&= \begin{bmatrix} \cos(\beta) \\ \sin(\beta) \end{bmatrix}\dot{r} + \begin{bmatrix} -r\sin(\beta) \\ r\cos(\beta) \end{bmatrix}\dot{\beta} && \text{(A.5)}\\
&= \dot{r}\mathbf{e}_r + r\dot{\beta}\mathbf{e}_\beta && \text{(A.6)}\\
&= \begin{bmatrix} \dot{r} \\ r\dot{\beta} \end{bmatrix}_{\mathrm{L}} && \text{(A.7)}
\end{aligned}$$

The second time derivative of the position vector, representing acceleration, is:

$$\ddot{\mathbf{p}} = \frac{d^2\mathbf{p}}{dt^2} = \frac{d}{dt}\left(\frac{d\mathbf{p}}{dt}\right) \tag{A.8}$$

$$= \frac{d}{dt}\left(\frac{\partial \mathbf{p}}{\partial r}\dot{r} + \frac{\partial \mathbf{p}}{\partial \beta}\dot{\beta}\right) \tag{A.9}$$

$$= \frac{\partial \mathbf{p}}{\partial r}\ddot{r} + \frac{\partial \mathbf{p}}{\partial \beta}\ddot{\beta} + \frac{\partial^2 \mathbf{p}}{\partial r^2}\dot{r}^2 + \frac{\partial^2 \mathbf{p}}{\partial \beta^2}\dot{\beta}^2 + 2\frac{\partial^2 \mathbf{p}}{\partial r \partial \beta}\dot{r}\dot{\beta} \tag{A.10}$$

$$= \begin{bmatrix}\cos(\beta)\\ \sin(\beta)\end{bmatrix}\ddot{r} + \begin{bmatrix}-r\sin(\beta)\\ r\cos(\beta)\end{bmatrix}\ddot{\beta} + \begin{bmatrix}0\\ 0\end{bmatrix}\dot{r}^2 + \begin{bmatrix}-r\cos(\beta)\\ -r\sin(\beta)\end{bmatrix}\dot{\beta}^2 + \begin{bmatrix}-\sin(\beta)\\ \cos(\beta)\end{bmatrix}2\dot{r}\dot{\beta} \tag{A.11}$$

$$= \left(\ddot{r} - r(\dot{\beta})^2\right)\mathbf{e}_r + (r\ddot{\beta} + 2\dot{r}\dot{\beta})\mathbf{e}_\beta \tag{A.12}$$

$$= \begin{bmatrix}\ddot{r} - r(\dot{\beta})^2\\ r\ddot{\beta} + 2\dot{r}\dot{\beta}\end{bmatrix}_{\mathrm{L}} \tag{A.13}$$

$$= \begin{bmatrix}\ddot{r}\\ r\ddot{\phi}\cos(\beta)\\ r\ddot{\beta}\end{bmatrix}_{\mathrm{L}} \tag{A.14}$$

A.2 3D Model

The position vector $\mathbf{p}$ in a 3D space is given by:

$$\mathbf{p} = \begin{bmatrix}r\cos(\beta)\cos(\phi)\\ r\cos(\beta)\sin(\phi)\\ r\sin(\beta)\end{bmatrix} \tag{A.15}$$

The time derivative of the position vector, representing velocity, is:

$$\dot{\mathbf{p}} = \frac{d\mathbf{p}}{dt} \tag{A.16}$$

$$= \frac{\partial \mathbf{p}}{\partial r}\frac{dr}{dt} + \frac{\partial \mathbf{p}}{\partial \phi}\frac{d\phi}{dt} + \frac{\partial \mathbf{p}}{\partial \beta}\frac{d\beta}{dt} \tag{A.17}$$

$$= \frac{\partial \mathbf{p}}{\partial r}\dot{r} + \frac{\partial \mathbf{p}}{\partial \phi}\dot{\phi} + \frac{\partial \mathbf{p}}{\partial \beta}\dot{\beta} \tag{A.18}$$

$$= \begin{bmatrix}\cos(\beta)\cos(\phi)\\ \cos(\beta)\sin(\phi)\\ \sin(\beta)\end{bmatrix}\dot{r} + \begin{bmatrix}-r\cos(\beta)\sin(\phi)\\ r\cos(\beta)\cos(\phi)\\ 0\end{bmatrix}\dot{\phi} + \begin{bmatrix}-r\sin(\beta)\cos(\phi)\\ -r\sin(\beta)\sin(\phi)\\ r\cos(\beta)\end{bmatrix}\dot{\beta} \tag{A.19}$$

$$= \dot{\mathbf{r}}\mathbf{e}_r + r\dot{\phi}\cos(\beta)\mathbf{e}_\phi + r\dot{\beta}\mathbf{e}_\beta \tag{A.20}$$

$$= \begin{bmatrix}\dot{r}\\ r\dot{\phi}\cos(\beta)\\ r\dot{\beta}\end{bmatrix}_{\mathrm{L}} \tag{A.21}$$

The second time derivative of the position vector, representing acceleration, is:

$$\ddot{\mathbf{p}} = \frac{d^2\mathbf{p}}{dt^2} = \frac{d}{dt}\left(\frac{d\mathbf{p}}{dt}\right) \tag{A.22}$$

$$= \frac{d}{dt}\left(\frac{\partial \mathbf{p}}{\partial r}\dot{r} + \frac{\partial \mathbf{p}}{\partial \phi}\dot{\phi} + \frac{\partial \mathbf{p}}{\partial \beta}\dot{\beta}\right) \tag{A.23}$$

$$= \frac{\partial \mathbf{p}}{\partial r}\ddot{r} + \frac{\partial \mathbf{p}}{\partial \phi}\ddot{\phi} + \frac{\partial \mathbf{p}}{\partial \beta}\ddot{\beta} + \frac{\partial^2 \mathbf{p}}{\partial r^2}\dot{r}^2 + \frac{\partial^2 \mathbf{p}}{\partial \phi^2}\dot{\phi}^2 + \frac{\partial^2 \mathbf{p}}{\partial \beta^2}\dot{\beta}^2 \tag{A.24}$$

$$+ 2\frac{\partial^2 \mathbf{p}}{\partial r \partial \phi}\dot{r}\dot{\phi} + 2\frac{\partial^2 \mathbf{p}}{\partial \phi \partial \beta}\dot{\phi}\dot{\beta} + 2\frac{\partial^2 \mathbf{p}}{\partial \beta \partial r}\dot{\beta}\dot{r} \tag{A.25}$$

$$= \begin{bmatrix} \cos(\beta)\cos(\phi) \\ \cos(\beta)\sin(\phi) \\ \sin(\beta) \end{bmatrix}\ddot{r} + \begin{bmatrix} -r\cos(\beta)\sin(\phi) \\ r\cos(\beta)\cos(\phi) \\ 0 \end{bmatrix}\ddot{\phi} + \begin{bmatrix} -r\sin(\beta)\cos(\phi) \\ -r\sin(\beta)\sin(\phi) \\ r\cos(\beta) \end{bmatrix}\ddot{\beta} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\dot{r}^2 \tag{A.26}$$

$$+ \begin{bmatrix} -r\cos(\beta)\cos(\phi) \\ -r\cos(\beta)\sin(\phi) \\ 0 \end{bmatrix}\dot{\phi}^2 + \begin{bmatrix} -r\cos(\beta)\cos(\phi) \\ -r\cos(\beta)\sin(\phi) \\ -r\sin(\beta) \end{bmatrix}\dot{\beta}^2 + \begin{bmatrix} -\cos(\beta)\sin(\phi) \\ \cos(\beta)\cos(\phi) \\ 0 \end{bmatrix} 2\dot{r}\dot{\phi} \tag{A.27}$$

$$+ \begin{bmatrix} r\sin(\beta)\sin(\phi) \\ -r\sin(\beta)\cos(\phi) \\ 0 \end{bmatrix} 2\dot{\phi}\dot{\beta} + \begin{bmatrix} -\sin(\beta)\cos(\phi) \\ \sin(\beta)\sin(\phi) \\ \cos(\beta) \end{bmatrix} 2\dot{\beta}\dot{r} \tag{A.28}$$

$$= \begin{bmatrix} \ddot{r} - r\dot{\beta}^2 - r\dot{\phi}^2\left(1 - \sin^2(\beta)\right) \\ r\ddot{\phi}\cos(\beta) + 2\dot{r}\dot{\phi}\cos(\beta) - 2r\dot{\phi}\dot{\beta}\sin(\beta) \\ r\ddot{\beta} + 2\dot{r}\dot{\beta} + r\dot{\phi}^2\cos(\beta)\sin(\beta) \end{bmatrix}_{\mathrm{L}} \tag{A.29}$$

$$= \begin{bmatrix} \ddot{r} - r\dot{\beta}^2 - r\dot{\phi}^2\cos^2(\beta) \\ r\ddot{\phi}\cos(\beta) + 2\dot{r}\dot{\phi}\cos(\beta) - 2r\dot{\phi}\dot{\beta}\sin(\beta) \\ r\ddot{\beta} + 2\dot{r}\dot{\beta} + r\dot{\phi}^2\cos(\beta)\sin(\beta) \end{bmatrix}_{\mathrm{V}} \tag{A.30}$$

$$= \begin{bmatrix} \ddot{r} \\ r\ddot{\phi}\cos(\beta) \\ r\ddot{\beta} \end{bmatrix}_{\mathrm{L}} + \underbrace{\begin{bmatrix} -r\dot{\beta}^2 - r\dot{\phi}^2\cos^2(\beta) \\ 2\dot{r}\dot{\phi}\cos(\beta) - 2r\dot{\phi}\dot{\beta}\sin(\beta) \\ 2\dot{r}\dot{\beta} + r\dot{\phi}^2\cos(\beta)\sin(\beta) \end{bmatrix}_{\mathrm{L}}}_{-\frac{1}{m}\mathbf{F}_{\mathrm{inert}}} \tag{A.31}$$

Appendix B

Appendix B

Auxiliary Equations

The scripts for the simulations are derived in this chapter.

```
#!/usr/bin/python3

from PyQt5.QtWidgets import QApplication, QMainWindow, QPushButton, QMessageBox, QVBoxLayout, QWidget
import awebox as awe
from upwind_settings import set_upwind_settings
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.colors as mcolors
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.collections import LineCollection

def save_figure_with_prefix(fig, filename):
    fig.savefig(f'test3_{filename}.png')

def main():
    # Indicate desired system architecture
    options = {}
    options['user_options.system_model.architecture'] = {1:0}
    options = set_upwind_settings(options)

    # Indicate desired operation mode
    options['user_options.trajectory.type'] = 'power_cycle'
    options['user_options.trajectory.system_type'] = 'lift_mode'
    options['user_options.trajectory.lift_mode.windings'] = 4  # Increase the number of windings

    # Adjust the initial and final tether lengths within the options
    initial_tether_length = 50.0  # Initial tether length in meters
    final_tether_length = 50.0   # Final tether length in meters
    options['solver.initialization.l_t'] = initial_tether_length
    options['model.system_bounds.x.l_t'][0] = initial_tether_length
    options['model.system_bounds.x.l_t'][1] = final_tether_length

    # Set the initial conditions for kite's state variables
    options['solver.initialization.groundspeed'] = 15.0
    options['solver.initialization.inclination_deg'] = 45.0
    options['solver.initialization.cone_deg'] = 15.0

    # Indicate desired environment
    options['params.wind.z_ref'] = 100.0
    options['params.wind.power_wind.exp_ref'] = 0.15
    options['user_options.wind.model'] = 'power'
    options['user_options.wind.u_ref'] = 10.0

    # Indicate numerical NLP details
    options['nlp.n_k'] = 500  # Increase the number of intervals for 1-minute simulation
```

```
    options['nlp.collocation.u_param'] = 'zoh'
    options['user_options.trajectory.lift_mode.phase_fix'] = 'simple'
    options['solver.linear_solver'] = 'ma57'  # if HSL is installed, otherwise 'mumps'

    # Build and optimize the NLP (trial)
    trial = awe.Trial(options, 'Ampyx_AP2')
    trial.build()
    trial.optimize()

    # Draw some of the pre-coded plots for analysis
    trial.plot(['states', 'controls', 'constraints', 'quad'])

    # Extract information from the solution for independent plotting or post-processing
    plot_dict = trial.visualization.plot_dict
    outputs = plot_dict['outputs']
    time = plot_dict['time_grids']['ip']
    avg_power = plot_dict['power_and_performance']['avg_power'] / 1e3

    print('=====================================')
    print(f'Average power: {avg_power} kW')
    print('=====================================')

    # Debugging: Print available keys
    print(plot_dict['x'].keys())

    # Use the correct keys for the kite's position
    position = plot_dict['x']['q10']  # Assuming q10 represents the position state vector
    x_pos = position[0]
    y_pos = position[1]
    z_pos = position[2]

    min_z_index = np.argmin(z_pos)
    min_point = (x_pos[min_z_index], y_pos[min_z_index], z_pos[min_z_index])
    print(f'Lowest point of the trajectory: {min_point}')

     # Find all local minima in the trajectory (lowest z-coordinates)
    local_min_indices = (np.diff(np.sign(np.diff(z_pos))) > 0).nonzero()[0] + 1
    local_min_points = [(x_pos[i], y_pos[i], z_pos[i]) for i in local_min_indices]

    print('Local minima of the trajectory:')
    for point in local_min_points:
        print(point)

    # Save the second local minimum point if it exists
    one_min_point = local_min_points[0] if len(local_min_points) > 0 else None
    second_min_point = local_min_points[1] if len(local_min_points) > 1 else None
    three_min_point = local_min_points[2] if len(local_min_points) > 2 else None
    four_min_point = local_min_points[3] if len(local_min_points) > 3 else None

    # Plot with a single color
    fig1 = plt.figure()
    ax = fig1.add_subplot(111, projection='3d')

    # Choose a single color for the trajectory
    single_color = 'blue'

    # Plot the trajectory with the chosen single color
    for i in range(len(x_pos) - 1):
        ax.plot(x_pos[i:i+2], y_pos[i:i+2], z_pos[i:i+2], color=single_color)

    ax.scatter(x[0], y[0], z[0], color='red', label='Start', s=100)  # Start point at z = 20
    ax.scatter(x[final], y[final], z[final], color='green', label='End', s=50)  # End point near z = 20
    ax.set_xlabel('X [m]')
    ax.set_ylabel('Y [m]')
    ax.set_zlabel('Z [m]')
    ax.legend()
    plt.show()


if __name__ == '__main__':
    main()


#!usr/bin/python3
```

```
def set_kite_simulation_settings(options):
    # Simulation parameters
    options['simulation.wind_velocity'] = 5.0  # Wind velocity in m/s
    options['simulation.gravity'] = 9.8        # Gravitational acceleration in m/s^2
    options['simulation.mass'] = 1.0           # Mass in kg
    options['simulation.time_horizon'] = 10.0  # Total simulation time in seconds
    options['simulation.control_intervals'] = 250  # Number of control intervals
    options['simulation.tolerance_radius'] = 1.0  # Tolerance radius for final state [m]

    # Tether parameters
    options['tether.cd'] = 1.2
    options['tether.rho'] = 0.0046 * 4 / (np.pi * 0.002**2)
    options['tether.diameter'] = 2e-3  # Tether diameter in m
    options['tether.use_wound_tether'] = False
    options['tether.control_variable'] = 'ddl_t'

    # Flight envelope
    options['flight.airspeed_limits'] = np.array([10.0, 32.0])  # [m/s]
    options['flight.beta_max_deg'] = 20  # Max sideslip angle in degrees
    options['flight.beta_min_deg'] = -20  # Min sideslip angle in degrees
    options['flight.alpha_max_deg'] = 9   # Max angle of attack in degrees
    options['flight.alpha_min_deg'] = -6  # Min angle of attack in degrees

    # Variable bounds
    options['bounds.l_t'] = [10.0, 700.0]  # Tether length bounds [m]
    options['bounds.dl_t'] = [-15.0, 20.0]  # Tether reel-out velocity bounds [m/s]
    options['bounds.ddl_t_max'] = 2.4       # Max tether acceleration [m/s^2]

    # Initialization settings
    options['initial.groundspeed'] = 15
    options['initial.inclination_deg'] = 45
    options['initial.cone_deg'] = 15
    options['initial.l_t'] = 200

    return options

def setup_optimization_problem():
    # Initialize options dictionary
    options = {}
    set_kite_simulation_settings(options)

    # Define decision variables and optimization problem
    opti = ca.Opti()  # Optimization problem setup

    # State trajectory
    X = opti.variable(3, options['simulation.control_intervals'] + 1)  # [r, beta, betad]
    r = X[0, :]      # Radius
    beta = X[1, :]   # Elevation angle
    betad = X[2, :]  # Rate of change of elevation angle

    # Control trajectory
    U = opti.variable(2, options['simulation.control_intervals'])  # [vt, alpha]
    vt = U[0, :]    # Tether reel-out velocity
    alpha = U[1, :] # Angle of attack

    # Define the system dynamics
    def f(x, u):
        return ca.vertcat(
            u[0],  # dr = vt
            x[2],  # d_beta = betad
            (1 / (options['simulation.mass'] * x[0])) * (
                -options['simulation.gravity'] * options['simulation.mass'] * ca.cos(x[1])
                + Faer_beta(u[1], options['simulation.wind_velocity'], x[0], u[0], x[1], x[2], options['flight.airspeed_limits'], options['simulation.cont
                - (2 * options['simulation.mass'] * u[0] * x[2])
            )
        )

    # Initial and final conditions
    r0, rf = 10, 30
    beta0, betaf = np.pi/6, np.pi/4
    betad0, betadf = 0, 0

    # State bounds
```

```
rmin, rmax = 10, 500
betamin, betamax = np.pi/9, np.pi/2

# Define Runge-Kutta integration and cost function
J = 0  # Initialize the cost function
dt = options['simulation.time_horizon'] / options['simulation.control_intervals']  # Time step

for k in range(options['simulation.control_intervals']):
    k1 = f(X[:, k], U[:, k])
    k2 = f(X[:, k] + dt/2 * k1, U[:, k])
    k3 = f(X[:, k] + dt/2 * k2, U[:, k])
    k4 = f(X[:, k] + dt * k3, U[:, k])
    T_value = tension(options['simulation.gravity'], options['simulation.mass'], beta[k], alpha[k], r[k], betad[k], vt[k], options['simulation.wind_ve
    J += (vt[k] * T_value) * dt  # Accumulate cost
    x_next = X[:, k] + dt/6 * (k1 + 2*k2 + 2*k3 + k4)
    opti.subject_to(X[:, k+1] == x_next)  # Close the gaps

opti.minimize(-J)  # Minimize negative power for maximization

# Set up constraints
max_vt = options['simulation.wind_velocity']
max_alpha = np.pi / 2  # Maximum alpha in radians

for k in range(options['simulation.control_intervals']):
    opti.subject_to(-max_vt <= vt[k])  # Lower bound on vt
    opti.subject_to(vt[k] <= max_vt)   # Upper bound on vt
    opti.subject_to(-max_alpha <= alpha[k])  # Lower bound on alpha
    opti.subject_to(alpha[k] <= max_alpha)   # Upper bound on alpha

# Initial state constraints
opti.subject_to(r[0] == r0)
opti.subject_to(beta[0] == beta0)
opti.subject_to(betad[0] == betad0)

# Final state within tolerance
opti.subject_to((r[-1] - rf)**2 <= options['simulation.tolerance_radius']**2)

# State bounds
opti.subject_to(rmin <= r)  # Lower bound on r
opti.subject_to(r <= rmax)  # Upper bound on r
opti.subject_to(betamin <= beta)  # Lower bound on beta
opti.subject_to(beta <= betamax)  # Upper bound on beta

# Initial guesses
t_vals = np.linspace(0, options['simulation.time_horizon'], options['simulation.control_intervals'] + 1)
initial_r = np.linspace(r0, rf, options['simulation.control_intervals'] + 1)
initial_beta = np.linspace(beta0, betaf, options['simulation.control_intervals'] + 1)
initial_betad = np.linspace(betad0, betadf, options['simulation.control_intervals'] + 1)
initial_vt = 0.1 * np.ones(options['simulation.control_intervals'])  # Small initial tether velocity
initial_alpha = np.zeros(options['simulation.control_intervals'])    # Zero initial angle of attack

opti.set_initial(r, initial_r)
opti.set_initial(beta, initial_beta)
opti.set_initial(betad, initial_betad)
opti.set_initial(vt, initial_vt)
opti.set_initial(alpha, initial_alpha)

# Solver options
opts = {}
opts["ipopt.print_level"] = 5
opts["ipopt.tol"] = 1e-4
opts["ipopt.max_iter"] = 5000 # este parametro e ajustavel
opts["ipopt.mu_strategy"] = 'adaptive'
opts["ipopt.bound_relax_factor"] = 1e-5

opti.solver("ipopt", opts) # ipopt from older library - usar nova no script a partir do path correspondente

# Solve the problem
#try:
#    sol = opti.solve()
#    print("Optimization successful!")
#except Exception as e:
#    print("Error during optimization:")
```

```
    #    print(e)

    # Post-processing
    t = np.linspace(0, options['simulation.time_horizon'], options['simulation.control_intervals'] + 1)

    #if 'sol' in locals():
    #    # Plot radius
    #    plt.figure()
    #     plt.plot(t, sol.value(r))
    #    plt.axhline(rmin, color='r', linestyle='--', label='Minimum Radius')
    #    plt.axhline(rmax, color='r', linestyle='--', label='Maximum Radius')
    #    plt.xlabel('Time [s]')
    #    plt.ylabel('Radius [m]')
    #    plt.title('Radius vs. Time')
    #    plt.grid(True)
    #    plt.legend()
    #    plt.show()
    #
    #    # Plot elevation angle beta
    #    plt.figure()
    #    plt.plot(t, sol.value(beta))
    #    plt.axhline(betamin, color='r', linestyle='--', label='Minimum Beta')
    #    plt.axhline(betamax, color='r', linestyle='--', label='Maximum Beta')
    #    plt.xlabel('Time [s]')
    #    plt.ylabel('Elevation Angle [rad]')
    #    plt.title('Elevation Angle vs. Time')
    #    plt.grid(True)
    #    plt.legend()
    #    plt.show()
    #
    #    # Plot rate of change of elevation angle betad
    #    plt.figure()
    #    plt.plot(t, sol.value(betad))
    #    plt.xlabel('Time [s]')
    #    plt.ylabel('Rate of Change of Elevation Angle [rad/s]')
    #    plt.title('Rate of Change of Elevation Angle vs. Time')
    #    plt.grid(True)
    #    plt.show()
    #
    #    # Plot controls
    #    plt.figure()
    #    plt.step(t[:-1], sol.value(vt), 'r', label='Tether reel-out velocity')
    #    plt.step(t[:-1], sol.value(alpha), 'b', label='Angle of attack')
    #    plt.xlabel('Time [s]')
    #    plt.ylabel('Controls')
    #    plt.title('Controls vs. Time')
    #    plt.grid(True)
    #    plt.legend()
    #    plt.show()

def Faer(alpha, vw, r, vt, beta, betad, rho, A):
    va = np.sqrt((vw * np.cos(beta) - vt)**2 + (vw * np.sin(beta) + r * betad)**2)
    cD = 0.012 + 0.01 * alpha
    cL = 0.3 + 0.1 * alpha
    R90 = np.array([[0, -1], [1, 0]])
    va_vec = np.array([vw * np.cos(beta) - vt, vw * np.sin(beta) + r * betad])
    return 0.5 * rho * A * va * (cD * va_vec + cL * (R90 @ va_vec))

def Faer_beta(alpha, vw, r, vt, beta, betad, rho, A):
    Faer_val = Faer(alpha, vw, r, vt, beta, betad, rho, A)
    return Faer_val[1]

def tension(g, m, beta, alpha, r, betad, vt, vw, rho, A):
    F_r = Faer_r(alpha, vw, r, vt, beta, betad, rho, A)
    return -g * m * np.sin(beta) + F_r + m * r * betad**2

def Faer_r(alpha, vw, r, vt, beta, betad, rho, A):
    Faer_val = Faer(alpha, vw, r, vt, beta, betad, rho, A)
    return Faer_val[0]
```