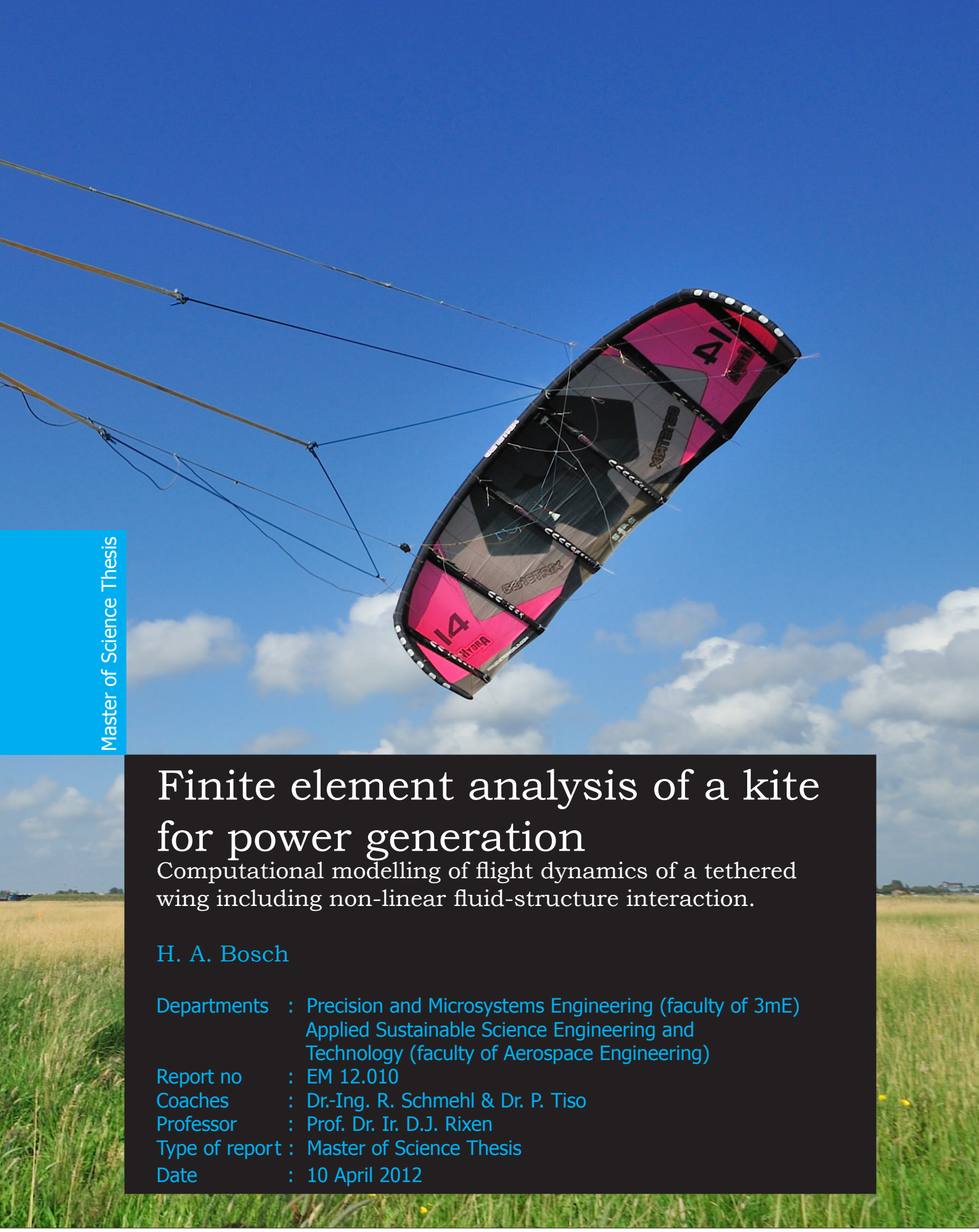


Master of Science Thesis

Finite element analysis of a kite for power generation

Computational modelling of flight dynamics of a tethered wing including non-linear fluid-structure interaction.

H. A. Bosch

Departments : Precision and Microsystems Engineering (faculty of 3mE)
Applied Sustainable Science Engineering and Technology (faculty of Aerospace Engineering)

Report no : EM 12.010
Coaches : Dr.-Ing. R. Schmehl & Dr. P. Tiso
Professor : Prof. Dr. Ir. D.J. Rixen
Type of report : Master of Science Thesis
Date : 10 April 2012

TU Delft Delft University of Technology

Challenge the future

Finite element analysis of a kite for power generation

Computational modelling of flight dynamics of a tethered wing including non-linear fluid-structure interaction.

Master of Science Thesis

For the degree of Master of Science in Mechanical Engineering at the Delft University of Technology

H.A. Bosch

April 2012

Faculty of Mechanical, Maritime and Materials Engineering (3mE)
Faculty of Aerospace Engineering
Delft University of Technology

Delft University of Technology
Departments of
Precision and Microsystems Engineering (PME) and
Applied Sustainable Science Engineering and Technology (ASSET)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Finite element analysis of a kite for power generation

by

H.A. Bosch

in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

Dated: April 2012

Supervisor(s):

Prof. Dr. D.J. Rixen

Dr.-Ing. R. Schmehl

Dr. P. Tiso

Reader(s):

Dr. A. A. Zadpoor

Preface

Working on this Master thesis was an exciting time and will bring an end to a great time of studying Mechanical Engineering at the Delft University of Technology. From the start I was very interested in the topic of this thesis, because it is a subject I can really relate myself to. Working on solutions for the energy challenge that my generation faces is not an easy task, but a very important one. I would like to think that I was able to contribute to these developments with this thesis.

I am grateful that I had the opportunity to combine the knowledge of two research groups from different faculties: the Research group Engineering Dynamics from the Mechanical Engineering faculty and the Kite Power research group from the Applied Sustainable Science Engineering and Technology (ASSET) department from the Aerospace Engineering faculty. People in both departments really showed passion for what they are doing and that inspired me to give my best as well.

I would like to thank Prof. Dr. Wubbo J. Ockels for starting the kite powered energy concept, giving me the opportunity to do this interesting thesis. It was also an honour to be able to work under the supervision of Prof. Dr. Daniel J. Rixen and learn from his great expertise in the dynamics field. My special thanks goes to my two supervisors Dr. Paolo Tiso and Dr.-Ing Roland Schmehl. This thesis wouldn't be here without all the hours they put into guiding me, helping me to solve the problems and reviewing my work. I would like to thank Dr. Amir A. Zadpoor for reading this thesis and being part of my graduation committee.

I would also like to show my gratitude to everyone in the Kite Power group. I learned a lot from their kite experience and had many important discussions and brainstorms with almost everyone within the group. My thanks also goes to the Precision and Microsystems Engineering (PME) department and all the feedback they gave me during the monthly presentations. This helped me to solve difficult mechanical problems and improved my work.

My last, but most important thanks goes my parents, friends an family and all the support they gave me throughout this and the past years of my studies.

Delft, University of Technology — H.A. Bosch
April 2012

Summary

Airborne Wind Energy is a new field of research that focusses on sustainable electricity generation from wind with flying devices that use less material and weigh less than conventional wind turbines. They can also access wind at higher altitudes where the power density is much higher and the wind velocity more constant than close to the earth's surface.

The Delft University of Technology developed a concept with a flexible kite that flies at a high altitude and is connected to the ground with a long tether. The tether is reeled off a drum that is connected to a generator that produces energy while the kite flies crosswind through the air in a figure eight. The kite stops flying crosswind when it reaches its maximum altitude and starts to hover like a parachute so that it can be reeled in with a low amount of energy. This results in a cyclic process that will produce a net amount of energy.

A leading edge and several struts form the backbone of the kite and are made from inflatable beams. A thin canopy between them introduces the aerodynamic forces to the structure. The kite is connected with a set of bridle lines to a control pod that is connected to the tether. The control pod is used to steer and control the power in the kite

Realistic computer models of kites are necessary to optimize this system, develop controllers for a fully autonomous system, design better kites and get a better understanding of the dynamic behaviour of kites. Existing kite models either lack the needed realism to perform these studies by neglecting the flexible behaviour of the kite or are too computational intensive to use when they do include the full flexible behaviour.

Therefore this thesis introduces a new realistic and reduced approach to model flying flexible inflatable tube kites used in airborne wind energy systems. The approach should result in a realistic model that represents the global dynamics and deformation modes of the kite, can be steered with real steering inputs (shortening and lengthening of the steering lines) and should be derived from physical principles. The model should be reduced to make it fast by neglecting unimportant local effects without losing the essential dynamics and it should be set-up so that it can be even further reduced with new model reduction techniques.

The system is split in three parts in the new approach: the kite, the aerodynamics and the bridle lines with the tether.

The kite is modelled with the non-linear finite element method to stay close to its physical properties and represent its full non-linear flexible behaviour. The inflatable beams in the leading edge and struts are modelled with regular beam elements that represent the behaviour to reduce the amount of degrees of freedom drastically. The canopy is modelled with shell elements and uses a coarse mesh to reduce the computation time without losing essential deformation modes.

The aerodynamic model of Breukels [7] is used to describe the distributed aerodynamic forces exerted on the kite depending on its deformation. The model is based on the assumption that the kite can be divided into a finite number of two dimensional wing sections. The aerodynamic properties are determined for each wing section from a set of two dimension CFD experiments and distributed over the wing section using a set of weight factors.

The finite element kite model and aerodynamic model form a fluid-structure-interaction problem together that needs to be solved iteratively. A solving algorithm is introduced that splits the structural and aerodynamic convergence and uses a load stepping algorithm to stabilize and speed-up convergence.

The bridles and tether are modelled dynamically as simple spring-dampers using multi-body dynamic methods. The end of the bridles are connected to the kite at the bridle attachment points

The most important reduction principle is the assumption that the kite reacts quasi-static to the aerodynamic forces, because the inertia of the kite are very small compared to the aerodynamic forces. Therefore the local inertia of the kite are neglected when solving the finite element equations and the dynamic deformations of the kite can be approximated by a sequence of static solutions. The quasi-static fluid-structure-interaction problem returns forces that are exerted on the end points of the bridles in the dynamic simulation. These forces are assumed to remain constant during the period of a time step, eliminating the need to solve the fluid-structure-action problem in the dynamic differential equations. This method greatly reduces the amount of computation time, because the time integration only has to be done for the small number of degrees of freedom in the dynamic model and the fluid-structure-interaction problem only needs to be solved once in every time-step instead of multiple times in the time integration algorithm.

All system components were implemented in MATLAB and a controller was built to fly several figure eight tests with the model. Results show that the modelling approach leads to a fast and realistic model. A steering input results in dynamics and a real deformation that causes the kite to yaw comparable to real kites and other models. It can be concluded that all the assumptions were valid and led to a model that captures the kite behaviour realistically. The model is a factor 25-30 slower than real time, which is very fast considering the complexity of the calculations and that it was implemented in a non compiled language. The clearly visible distinct deformation modes and non-linear force model make it also a suitable and interesting candidate for further new model reduction techniques. The aerodynamic model has some shortcomings resulting in an overestimation of the lift over drag ratio and is considered to be the main source of uncertainties in the whole model. Replacement with a better model should further improve the model and make it possible to do an extensive validation study.

Concluding, the new proposed approach is successful: fast and realistic, flexible in its use, able to model all type of kites, a good candidate for further reduction and can be used for controller design and optimizing studies.

Table of Contents

Chapter 1

Introduction: kite power as solution to the world energy challenge

1-1 The energy challenge

Energy is one of the most important factors in our daily life nowadays and we are using more and more of it. According to the International Energy Outlook 2011 [1] the world energy consumption will increase from 505 quadrillion Btu in 2008 to 770 quadrillion Btu in 2035, an increase of 53 percent. The question that will keep humanity busy for the next decades will be how to obtain a sustainable solution to provide enough energy for the world demand.

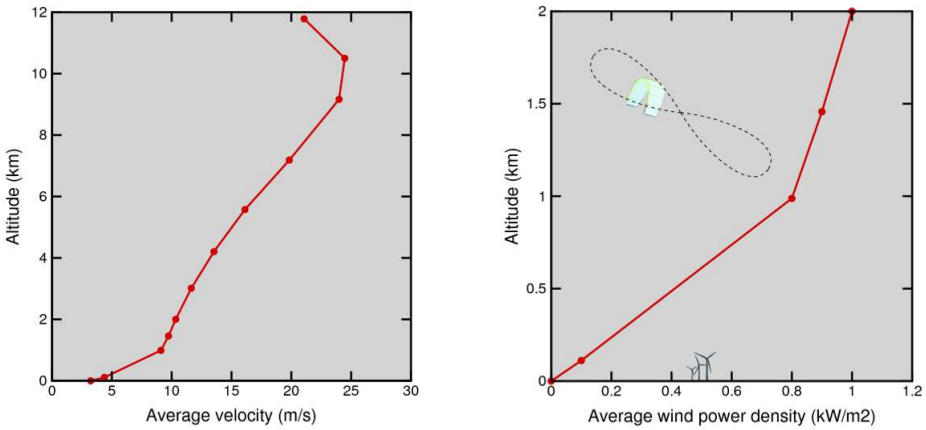
Electric energy forms a large part of this consumption and the demand will increase by 2.3 percent each year from 2008 to 2035, leading to a total increase of 84 percent from 19.1 trillion kilowatthours in 2008 to 35.2 kilowatthours in 2035.

Renewable electric energy will be the fastest growing source of electricity generation between 2008 and 2035 with an annual growth of 3.1 percent. In 2008 renewables accounted for 19 percent of the total electricity generation, with 3.7 trillion kilowatthours.

Wind energy production has grown very fast over the past decade, from 18 gigawatts of net installed capacity at the end of 2000 to 121 gigawatts at the end of 2008. This growth will continue: 27 percent of the 4.6 trillion kilowatthours growth in renewable electric energy from 2008 to 2035 will be due to wind energy. In 2008 the wind energy counted for 5.7 percent of the total renewable electricity generation with 210 billion kilowatthours and will have an average annual growth of 7.5 percent between 2008 and 2035.
Clearly wind energy plays an important role in the future of electric energy, but it has two problems. Firstly it can't compete economically with fossil fuels at the moment. The operating costs of wind turbines are much lower than the operation costs for fossil fuel-fired power plants, but the high construction costs make the total cost per kilowatthour higher. Secondly wind energy is an intermittent technology that is only available when there is wind. New storage techniques can be helpful to store a surplus of wind energy when there is much wind

that can be used in periods with a shortage of wind, but only a limited amount of energy can be stored.

Airborne Wind Energy (AWE) is new field of research and focusses on electricity generation with flying devices. These devices use a far less material and weigh much less than conventional wind turbines, making them cheaper. Another advantage is that they can access wind at higher altitudes. Figure 1-1 shows that the wind speed and wind power density at high altitude is much higher than at the operating altitude of wind turbines (maximum of 200 meters) and is a potential source of much sustainable energy. Another advantage of high altitude wind is that the wind is more constant than at low altitude. The boundary effect of the earth slows down the wind close to the surface. These factors show a large advantage of airborne wind energy over conventional wind energy: cheaper and access to a more constant and denser power source.
Over the past decade the amount of research in this field has grown significantly and several new concepts have been developed, but much more research is needed before these systems are technically mature and can compete economically with other energy generation methods.



(a) Wind speed at high altitude.

(b) Wind power density at high altitudes.

Figure 1-1: High altitude wind power in The Netherlands.

Kite powered energy is one of these new promising concepts. A large kite as depicted in Figure 1-2 is connected to the ground with a tether. The tether is reeled off a drum that is connected to a generator that produces energy while flying the kite crosswind through the air. The kite stops flying crosswind when it reaches its maximum altitude and starts to hover like a parachute so that it can be reeled in with a low amount of energy. This results in a cyclic process that will produce a net amount of energy as depicted in Figure 2-1). Besides being able to easily reach high altitudes, an advantage of the system is the low construction costs.

Many exciting research questions need to be answered in the coming years to perfect this system and be able to make it a commercial competing product. This thesis will focus on using kites as a solution to the world energy problem.

Figure 1-2: The kite power concept from the Delft University of Technology. Photo by Max Dereta.

1-2 Thesis objective

Reliable computer models of kites are necessary for further development of the kite power system. These models need to simulate the behaviour of a kite in a realistic way to further improve the design of the kites, to develop robust controllers for system automation and to do various system optimization studies. Kites are highly flexible structures that constantly interact with the changing aerodynamic forces, which makes it a very difficult problem to model. Existing kite models either lack the needed realism to perform these studies or are too computationally intensive to use when they do include the full flexible behaviour of the kite. Therefore the goal of this thesis is to *develop a new realistic and reduced approach to model flying flexible inflatable tube kites used in airborne wind energy systems.*

1-3 Thesis structure

To develop a new modelling approach, Chapter 2 first summarizes the current state of the research, explains the system in some more detail and states the research opportunities. Chapter 3 formulates the goal of this thesis and explains the important requirements followed by the description of the new modelling approach in Chapter 4. The new model exists of three main building blocks that are explained in the subsequent three chapters: the finite element kite model in Chapter 5, the aerodynamic model in Chapter 6 and the dynamic tether and bridle model in Chapter 7. The three components and a controller are integrated in one system in Chapter 8, followed by the results and discussion in Chapter 9.The final conclusions can be found in Chapter 10.

Chapter 2

Kite powered energy research

This chapter discusses the ideas behind kite powered energy. Section 2-1 describes the research performed at the Delft University of Technology, followed by the current state of the research on kite modelling in Section 2-2. The chapter ends with some concluding remarks in Section 2-3 that serve as a basis for the formulation of the goal and approach of this thesis in the next chapters.

2-1 The pumping crosswind power generating kite system

The idea for using a single crosswind flying kite powered system originates from the Laddermill concept that was developed by Ockels [37] in the 1990's where multiple kites are coupled together in one system. When serious research on this subject started at the Delft University of Technology around 2005 it was soon concluded that controlling and understanding a single kite was already difficult enough and the idea of using crosswind power [32] became the new focus. The first prototypes of the system were built around 2007 [29, 30, 28, 31, 27] and the first full scale prototype of a 20kW system was built by the Kite Power research group from the ASSET (Applied Sustainable Science Engineering and Technology) department in 2010 [12]. The last years were used to gain more insight in the system and current research focusses on optimizing the energy output, designing better kites and fully automating the system. This section describes the working principle behind the system in Section 2-1-1 and the several components in Section 2-1-2.

2-1-1 Crosswind power

The principle

Figure 2-1 illustrates the basic crosswind power concept. A kite is attached to a long tether that is connected to a power generator on the ground. The process consists of two phases: the generation phase and the retraction phase.

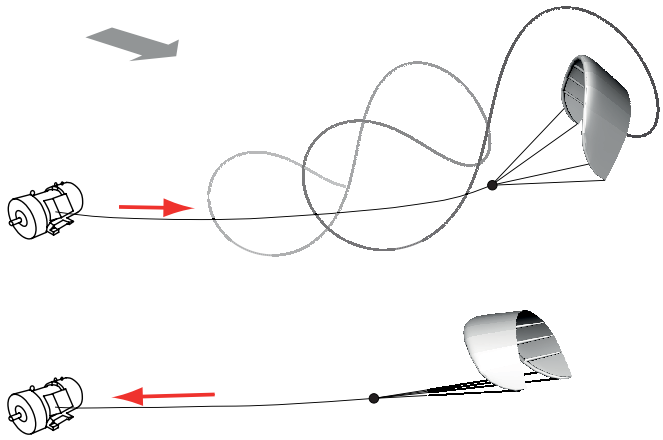


Figure 2-1: Working principle of the kite powered system. The kite is attached to a tether that is reeled off a drum that is connected to a generator on the ground while flying a crosswind figure eight pattern during the power generation phase. The kite is reeled in during the retraction phase with a minimum amount of energy.

The kites flies a figure eight pattern in the sky to obtain maximum lift forces when the system is in the generation phase. At the same time the tether is reeled off a drum that is connected to a generator which generates energy due to the revolving motion.

The retraction phase starts when the kite reaches a certain altitude. The kite is put in a de-powered mode where it generates almost no force on the tether by changing the pitch angle of the kite. The generator is used as a motor and reels in the kite to an altitude from where the generation phase starts again. The energy needed to reel in the kite during the retraction phase is only a fraction of the energy that is produced during the generation phase.

This results in a cyclic (pumping) process with a phase that generates energy and a phase that costs energy, but with a net positive effect. This is illustrated in Figure 2-2 where the mechanical power of a couple of cycles during test flight with the real system are shown.

Crosswind power

The kite flies crosswind figure eight patterns during the reel-out phase as depicted in Figure 2-1 and Figure 2-4. This crosswind principle was already described by Loyd [32] in 1980 and was further explored by Ockels [36] and Terink [46] and nicely summarized by de Groot [11].

The kite gains more speed when flying crosswind and generates an apparent wind speed that is higher than the normal wind speed. The lift force increases with the apparent wind speed and therefore the power generation as well. The generated power P can be expressed in terms of the tension force in the tether T_t and the reel-out speed of the tether V_t by $P = T_t \cdot V_t$. When the kite flies crosswind with a speed V_k it finds an equilibrium between the lift force L, the drag force D and the tether force T_t as depicted in Figure 2-3. Loyd [32] describes that the L/D ratio must be optimized to get a maximum power output, because the lift force varies quadratically with the L/D ratio. The maximum power generation is obtained when the reel out speed of the tether V_t is 1/3 of the wind speed V_w. The power production also depends on the cycle time and the tether length [52, 55]. Several optimization studies have shown that flying a figure eight is the most optimal trajectory [56, 18].

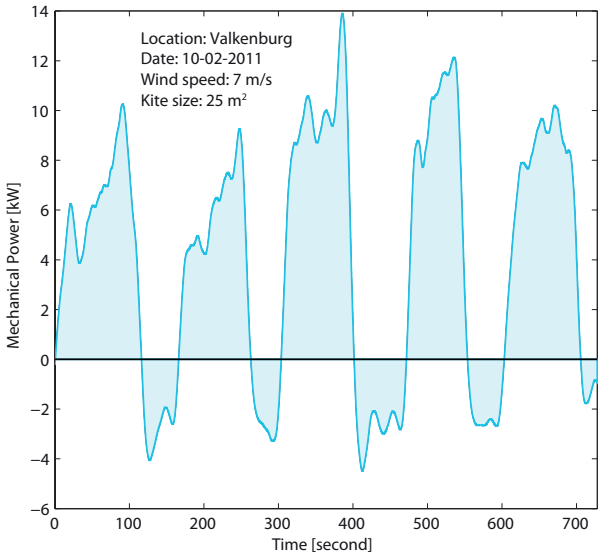


Figure 2-2: The power production of a couple of pumping cycles with the real system.

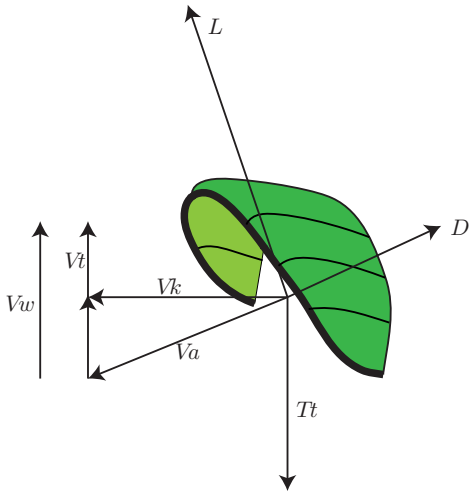


Figure 2-3: The force equilibrium between the lift force L, the drag force D and the tether force T_t in case of a horizontal cross-wind flight.

Figure 2-4: Figure eight manoeuvre during a cross wind flight.

Power scale

The most important question is how much energy the system can generate to what kilowatthour price. The current system has a nominal power rating of $20kW$ using a $25m^2$ kite, but studies have shown that it is possible to scale this system to the order of megawatts [31]. It is not easy to predict the kilowatthour price. The structure of a kite system is much

cheaper than for wind turbines and uses a more constant power source, but the maintenance costs for kite systems might be higher due to for instance the low durability of the materials. The future will tell how much the real costs will be.

Wind window

The earth surface creates a boundary effect that slows down the wind close to its surface. This is the reason for higher wind speeds and a more constant wind at high altitudes as was shown in Figure 1-1a. The wind velocity is never constant and always shows variations due to wind gusts and gaps. These effects are highly random and influence the power generation. Robustness of flying controllers is important to safely and optimally control the kite. de Groot [11] shows that the wind speed variation is about 10-20% in normal flight conditions. The maximum theoretical speed that the kite can obtain is approximately the L/D ratio times the wind speed as was shown by van den Heuvel [48], resulting in a much higher apparent wind speed.

Figure 2-5 shows that the flying domain of a kite exists of the quarter of a sphere. The amount of pluses and minuses shows how much power de kite generates when flying in that part of the domain. This helps to understand how the kite can be flown to produce the maximum amount of energy.

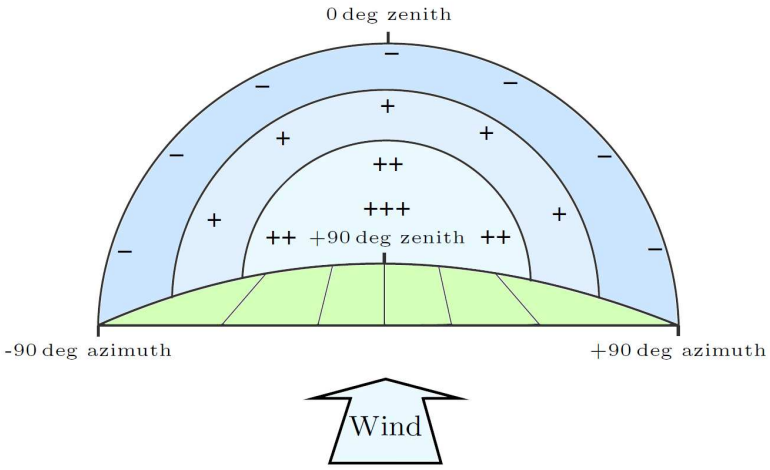
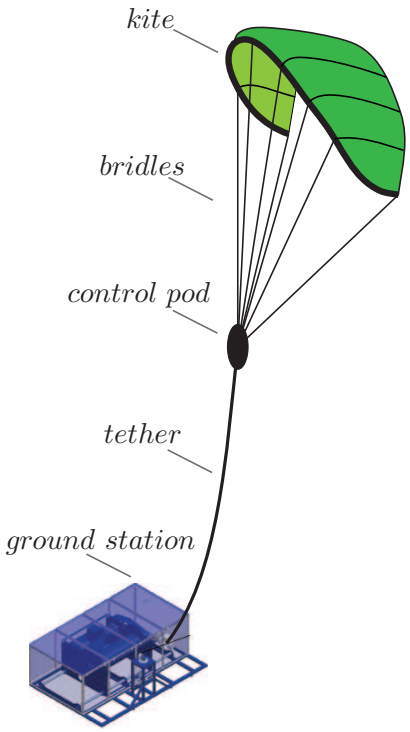


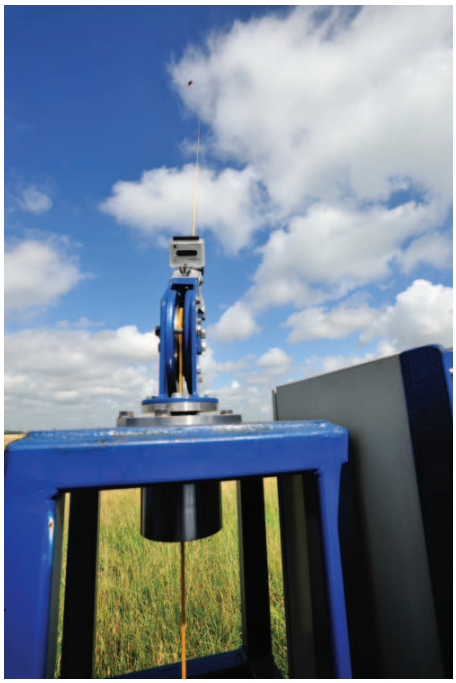
Figure 2-5: The wind window of a kite [18]. It can be flown in the space described by the quarter of a sphere, but experiences different power zones (+ / -).

2-1-2 System overview

The system consists of the core components as shown in Figure 2-6a. The kite is attached with a set of bridle lines to the control pod. The control pod contains the mechanism to steer the kite and to power or de-power it by changing the pitch angle and has a wireless connection with the ground station. A tether connects the control pod to the ground station. Figure 2-6b shows the real system that was built by the Delft University of Technology.



(a) The full system with all its components.



(b) An overview of the real system (photo from Max Dereta).

Figure 2-6: System overview

Kite

Numerous types of kites exist and have been tested for power generation. Figure 2-7 shows four different types. A division can be made between kites with and without inflatable parts. Both have their advantages, but the ASSET system currently uses kites with inflatable parts.

The kite exists of an inflatable leading edge and inflatable struts that form the structural backbone of the kite as can be seen in Figure 2-8. A very thin canopy forms the wing of the kite and generates the aerodynamic forces. Four lines are attached to the four corners of the kite to control it. The two front lines, called the power lines, handle most of the force. The two rear lines, called the steering lines, are used the steer the kite.

The kite is an extremely flexible structure as can be seen from Figure 2-10, which makes is hard to model it. This flexibility is necessary to give the kite its manoeuvrability. A difference in steering line lengths deforms the kite and results in a steering movement as will be further discussed in Chapter 9.

The length difference between the steering lines and the power lines determines the pitch angle of the kite, as shown in Figure 2-9. If the pitch angle increases, the lift force in the kite increases as well, called *powering* the kite. During the retraction phase the kite is *de-powered* which means that the pitch angle is decreased. Powering the kite too much will cause the kite to stall.

Small kites are very direct in their steering behaviour, while large kites are very slow in

(a) The Mutiny $25m^2$ inflatable tube kite.

(b) The Hydra $14m^2$inflatable tube kite.

(c) Peter Lynn foil kite.

(d) The C-shaped North Rhino $16m^2$ kite.

Figure 2-7: Different kite types.

their steering behaviour. This also depends on the wind speed. The performance of the kite is largely determined by its lift over drag ratio (L/D) as mentioned before. The L/D of arc-shaped kites ranges roughly from 4 to 7.

Bridles

Depending on the kite type, a set of bridles is used to attach the steering and power lines to the kite. These bridles are attached to the leading edge or struts at different points and keep the kite in an aerodynamic optimal shape during operation. Figure 2-11 shows an example of a bridled kite. Not all kites use bridles, the steering and power lines are directly connected to the four corners of C-shape kites for instance, as shown in Figure 2-7d.

Control pod

The control pod, as shown in Figure 2-12, contains all the necessary electronics to steer the kite and to (de-)power the kite. It has a wireless connection with the ground station and can

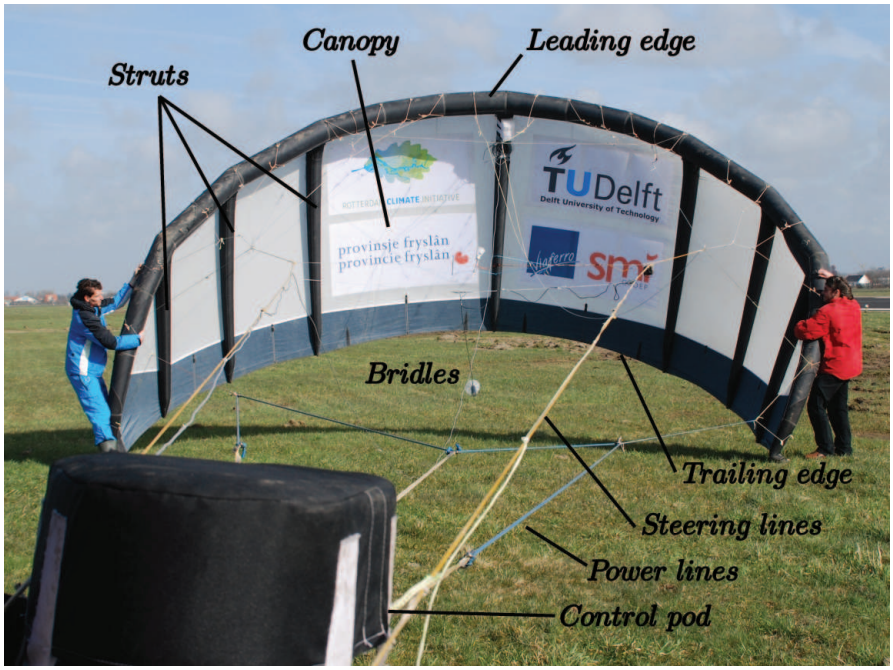


Figure 2-8: Naming conventions for an inflatable tube kite.

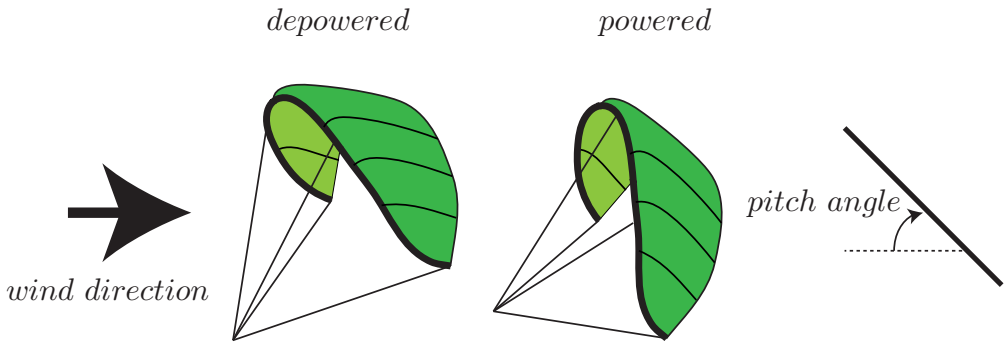


Figure 2-9: Powering and de-powering of the kite by changing the pitch angle resulting from a difference in lengths of the steering lines and power lines.

be used in manual control or automatic control. The power for all the components is provided by batteries. Currently they have to be recharged on the ground, but a small wind turbine on the control pod will recharge the batteries during the flight in the future.

Tether

The tether is made from Dyneema. Depending on the length and diameter of the tether it has an important role in the dynamics of the total system. The mass of the tether can become much higher than the mass of the kite when flying with long and thick tethers that are needed for large kites that generate high lift forces. The drag force on the tether increases as well with the tether length and diameter and becomes significant. Effects like sag of the cable also become important as can clearly be seen from Figure 2-13.

Figure 2-10: The extreme flexibility of a kite.

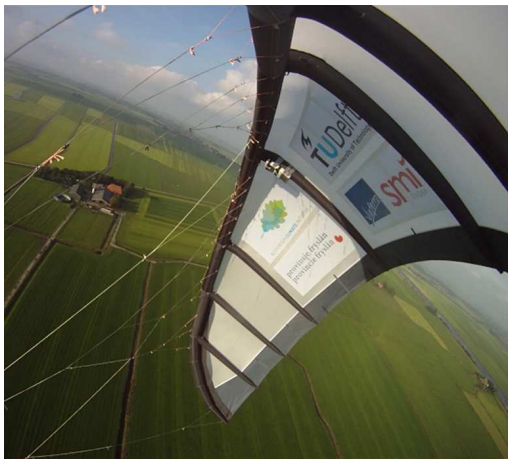
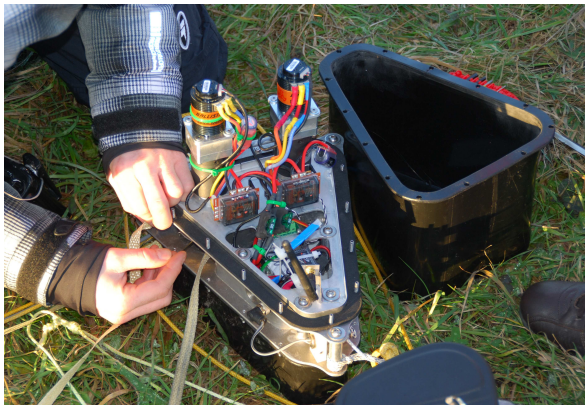


Figure 2-11: Bridle system of the Mutiny kite.



(a) Opened control pod.

(b) Hanging under the kite.

Figure 2-12: The control pod.

Ground station

The ground station as depicted in Figure 2-14 contains the drum, the generator and all the needed electronics to operate them. Multiple computer systems are used to control and monitor the kite when it is operational.

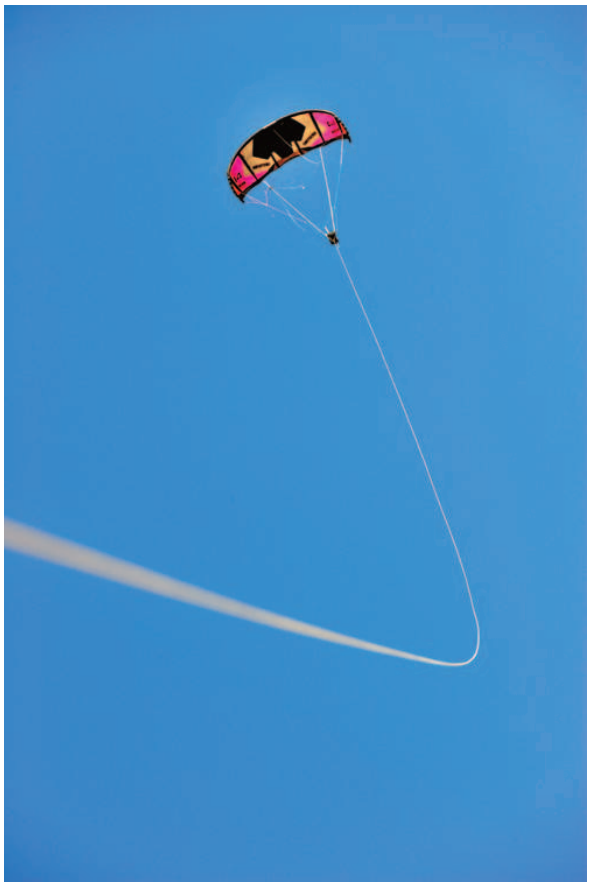


Figure 2-13: Extreme sag of the tether (photo by Max Dereta).

(a) The groundstation.

(b) The generator and drum.

Figure 2-14: The ground station.

2-2 Current research

Kite powered energy is a fairly new research field with many questions that still need to be answered. This makes it a very challenging and exciting subject. Some of the current research fields are the modelling of kites, tethers and ground stations, developing controllers, optimizing kite designs of flying trajectories, developing aerodynamic models, implementation strategies (air traffic, locations, legislation etc.) or studies on more durable kite materials.

This thesis focusses on the modelling part. A short overview will be given on the current state of the research of kite modelling in Section 2-2-1, tether modelling in Section 2-2-2 and aerodynamic modelling in Section 2-2-3

2-2-1 Kite models

Several kite models have been developed over the last years. Figure 2-15 gives an overview of all the models based on their complexity. Some of them have been verified with some basic experiments, but most of them haven't been verified at all. The models will be discussed in this section.

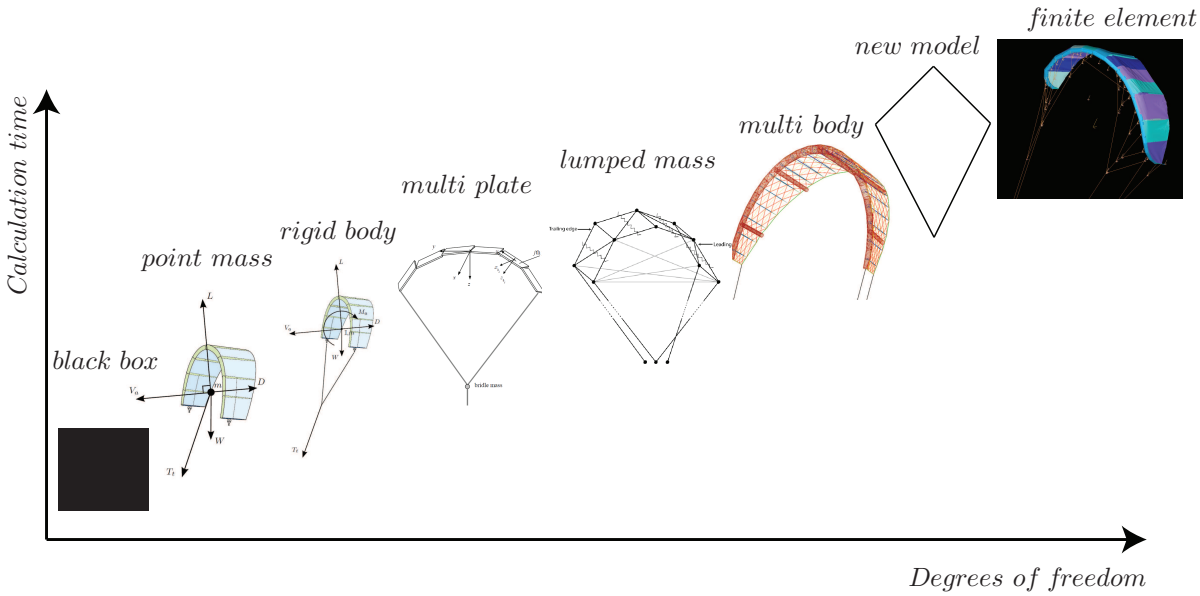


Figure 2-15: All the currently available kite models, sorted by their complexity.

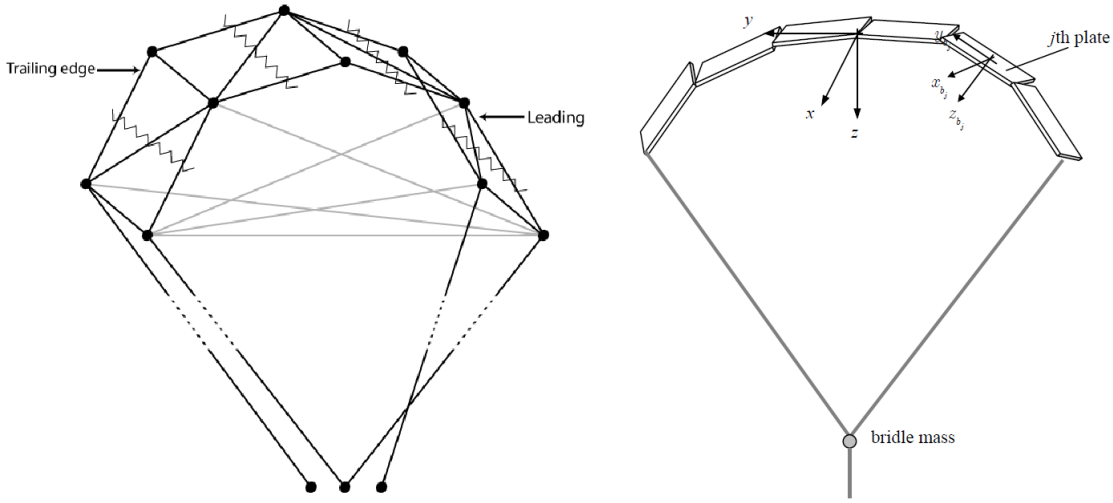
Black/grey box models are obtained with model identification techniques to fit mathematical equations to experimental data assuming no (black) or little (grey) knowledge of the physical system. One of these models is currently being developed at the Delft University of Technology and Erhard and Strauch [13] also developed a model without giving the details. The advantage of such models is that their behaviour directly represents the real experiments and that they are very fast. Nothing can be said however about the validity of the model when situations are modelled that did not occur in the measurement data. Furthermore no physical insight can be gained from them, nor can the effect of new kite designs be tested. The models are very suitable to compare to other models to judge their validity.

Points mass models are the simplest physical kite models available, but limited in their use. The kite is approximated as a point mass at the end of the tether on which lift, drag, tether and gravity forces work. The aerodynamic forces are determined from simple plate aerodynamic models. The flexibility of the kite and the accompanying dynamic effects are completely neglected and controlling is introduced by directly controlling the yaw and pitch angles to change the directions of the lift and drag forces. This makes them not very accurate and unsuitable to study kite behaviour or test new designs. They are mostly used for preliminary studies on trajectory optimization or system performances. These models have been developed by several persons, for instance by Williams et al. [56].

Rigid body models model the kite in the same manner as aircrafts are modelled. Williams et al. [56] designed a six degrees of freedom rigid model and included the moments of inertia of the kite. Houska [21] took a different approach and neglects the local moments of inertia of the kite by stating that they are much smaller than the combined global moment of inertia of the kite as point mass and the tether. He uses three degrees of freedom to describe the displacements of the kite, three degrees of freedom to describe the tether model and two degrees of freedom to simulate the control mechanism. He superimposes bending of the arced

shape of the kite as an additional state by introducing a second order differential equation. These models include more of the real dynamics of a kite than the point mass models, but still lack the flexibility of a real kite. de Groot [11] developed a rigid body model that was derived from the multi body model of Breukels [7] that will be discussed below. He concluded that it is very questionable whether it is possible to capture the dynamics of a kite realistically with a rigid body model.

A particle based model was developed by Furey and Harvey [17] as depicted in Figure 2-16a. The kite structure is described by a number of lumbed mass points that are connected with rigid rods and hinged together. The light grey lines are flexible constraints that are used to simulate the stiffness of the canopy and the struts. The zigzag lines represent aerodynamic slices for which the aerodynamic forces are calculated. This model is able to deform and therefore able to steer by pulling one of the steering lines, but the separate components do not flex and therefore it still lacks the real flexibility of a kite. Furthermore the needed properties to simulate the behaviour are quite artificial.



(a) The by Furey and Harvey [17] developed particle based kite model.

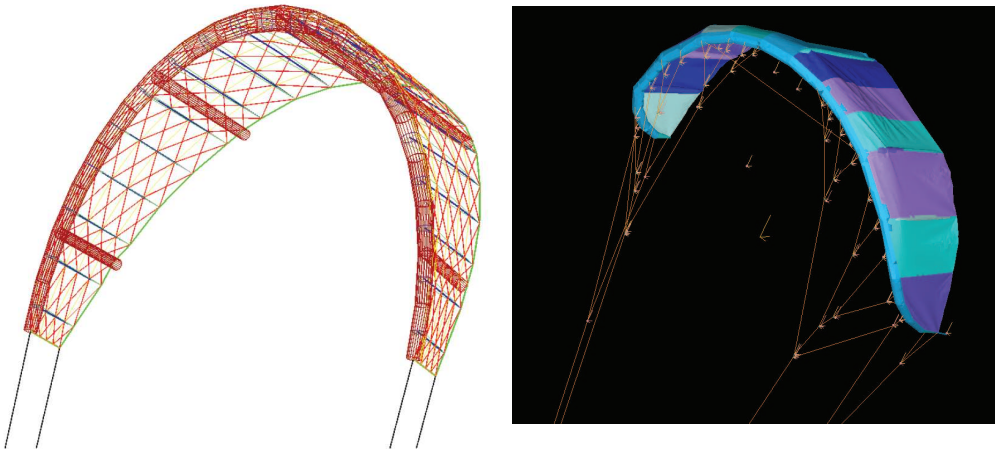
(b) The by Williams et al. [56] developed multi plate model.

Figure 2-16: More advanced kite models.

A multi-plate model was developed by Williams et al. [53] [56] and is shown in Figure 2-16b. The shape of the kite is approximated with a finite number of plates that are hinged together at the leading edge. The aerodynamic forces are determined for each plate separately from plate theory. The advantage of this model is that it includes some deformation modes of the kite, but the disadvantage is that it still lacks the realism of a real flexible kite model, since its elements are not able to flex. Furthermore this model has difficulties in finding an equilibrium position and therefore brings some computational issues.

A multi-body model developed by Breukels [7] was a completely new approach and he designed a whole framework to simulate flexible kites, as can be seen in Figure 2-17a. The multi body framework was used to define different building blocks that are used to build a complete kite. Rigid links are hinged together with spherical springs to form the leading edge and struts, and linear springs are used to simulate the canopy an tether. Parameters

were determined from various experiments to give the model realistic properties to simulate the correct beam stiffness and canopy stiffness. A new aerodynamic model was developed to calculate an distributed aerodynamic loads for the surface of the kite. This model simulates the flexibility of the kite in a very nice way and can use real steering inputs to steer te kite. The main problem is that it still relies on many artificial elements and parameters that have no direct physical meaning to simulate the physical behaviour of the kite.



(a) The by Breukels [7] developed multi body kite.

(b) The by Schwoll [43] developed finite element model developed with Madymo.

Figure 2-17: More advanced kite models.

A finite element model was developed by Schwoll [43] as shown in Figure 2-17b. The software package Madymo was used to build a very detailed model of a kite with more than 30.000 elements. This software was designed to simulate air bags and therefore has the nice option to inflate beams. The inflatable leading edge and canopy are modelled in a very realistic way. The great thing about the finite element approach is that the method is specially designed to simulate structural problems and intrinsically includes all the physical material properties in the model. There is no need for artificial parameters to simulate the physical behaviour of the kite, everything is already there. The main disadvantage of this model is that it is so detailed that the calculation times become very high. Furthermore, it needs a very sophisticated aerodynamic model that produces the distributed load on the canopy. Since this is not available, the kite doesn't fly yet and is only used form some structural experiments.

2-2-2 Tether models

Also several tether models were developed over the last years. The tether becomes an important part of the dynamics of the system when the tether becomes very long or thick due to the sag, mass and drag forces. Modelling the tether is not an easy task, due to the non-linear behaviour of the tether. Tether modelling is not the goal of this thesis and therefore only a very brief overview is given. A more detailed summary can be found in reference [11].

A rigid tether is the simplest modelling option and can only be used for simple simulations.

A linear spring damper represents the dynamics of a tether already quite well for a tether shorter than 100m and when there is enough tension in the tether to keep the sag small [5]. Stiffness and damping properties can be obtained by using Hook's law and the material properties.

Lumped masses connected together with springs can be used to approximate the tether dynamics and result in many more degrees of freedom than the earlier mentioned options, but gives a nice representation of the tether dynamics [5]. The disadvantage of this approach is that the springs will have a high stiffness which will result in a small time step in order to capture the high frequency behaviour and obtain a stable simulation.

Discretized inelastic rods coupled together with hinges are used by Breukels [7] and Williams et al. [54] to model the tether. Structural damping is introduced by adding dampers in the hinges. The problem with this approach is that very non-linear equations will arise or constraints have to be added to the system of equations.

2-2-3 Aerodynamic models

Various methods exist to model the aerodynamics for aircrafts. Most of those methods assume a rigid structure and are therefore not suitable for kites. Flexible kites deform under an aerodynamic load, resulting in a different aerodynamic load. This results in a aero-elastic problem that needs to be solved iteratively. The aerodynamic model that can be used depends on the chosen structural model of the kite. Flexible kite models need a more sophisticated aerodynamic model than rigid models. A short overview of available aerodynamic models that are used by the several structural kite models.

Plate theory is the simplest way to describe aerodynamic forces and assumes that a wing can be simulated as a rigid plate with a certain angle of attack and a lift and drag coefficient, from which the lift and drag force and airfoil moment can be determined. The airflows around kites are completely different from rigid wings and too turbulent to be approximated accurately with this method [7]. More details can be found in reference [34].

The vortex lattice method is a numerical Computational Fluid Dynamics (CFD) method that models a surface as an infinitely thin sheet of discrete vortices to compute lift and drag forces. It gives rough approximates of these forces and is often used in the early stages of aircraft design.

Aerodynamic derivatives are widely used in the dynamics of aircrafts. They are a fast way to compute the aerodynamic forces based on experience and the history of the states of the model. Details can be found in Mulder et al. [34]. This model is not applicable to flexible models.

The Breukels model was developed by Breukels [7] and used in his multi-body kite modelling approach. He performed two dimensional CFD simulations on airfoils with different physical properties (camber, angle of attack, thickness) and fitted functions through the results to approximate the lift, drag and aerodynamic moment curves for a two dimensional airfoil. The aerodynamics of the kite are approximated by dividing it into a finite number of aerodynamic sections in span-wise direction. For every section the aerodynamic forces are determined from the fitted functions and distributed over the surface by using a set of determined weight factors. This is the only aerodynamic model that takes the deformation of

the kite into account. The model has been validated to some extent, but is still a very rough approximation of the real aerodynamic situation. This model will be discussed in Chapter 6

Computational fluid dynamics (CFD) is the most detailed analysis that can be performed in the field of aerodynamics. The space around a structure is discretized with small volume elements and the fluid flows and pressures are calculated using fluid dynamics theory. This method calculates a pressure distribution of a surface over time. Computation times are low in the two dimension case, but become very high for the three dimensional analysis. This makes it impossible to use this method in dynamic simulations of the full kite model.

2-3 Conclusions and research opportunities

None of the discussed kite modelling approaches really satisfies, they are either too much simplified, not representing the real deformation modes of the kite or too computational intensive and none of them has successfully been verified

The simple models (point mass, rigid body) serve their purpose by giving a rough approximation of the kite behaviour. This is sufficient to do some preliminary optimization studies or to design the rough outline of a controller. The models lack the needed realism to test controllers for robustness, to see how kites behave in certain circumstances, to test new designs or to gain more insight in the steering and dynamic behaviour.

The intermediate models (particle based, multi-plate) try to incorporate some of the flexibility of the kites, but do that in an artificial way, neglecting the real flexibility.

The advanced models either are not able to fly and are too computational expensive to use (finite element model) or use many artificial fitted parameters and elements so that it is unclear how good it represents the reality (multi-body model).

It can be concluded that there is still need for better kite models that give a better representation of the highly flexible nature of the kite so that more insight can be gained in its (steering) behaviour and better controllers and kites can be developed. New models should have the right balance between calculation speed, amount of detail and physical validity.

Furthermore a good validation of all the currently available kite models is needed. The Kite Power research group at the Delft University of Technology is working on this problem.

Chapter 3

Thesis goal

The conclusions of Chapter 2 lead to the following formulation of the goal of this thesis. The goal of thesis is to:

---

*show a new **realistic** and **reduced** approach in the modelling of flying flexible inflatable tube kites used in airborne wind energy systems.*

The approach has to result in a **realistic** model.

- The global behaviour of the kite has to be represented realistically in the model. Local dynamic effects like fluttering of the canopy are not important.
- A real steering input of shortening a steering line should result in a deformation of the kite that creates a new force equilibrium and causes the roll and yaw motion. Shortening or lengthening both steering lines should result in (de-)powering of the kite.
- The model should be derived from physical principles and not depend on artificially introduced elements or many fitted parameters. This gives more insight in the physical properties and behaviour of the kite, the ability to use real material properties and naturally remains close to reality.

The approach has to result in a **reduced** model.

- The model should be as fast as possible to be a handy tool for controller design, performing optimizing studies and kite designers.
- The method needs to reduce the real kite to a simplified model, without losing the essential information to capture the global dynamic behaviour.
- The model must be built so that it can be further reduced by model reduction techniques as used in the finite element method field.

Two additional requirements are formulated.

- The new approach should be flexible so that it is easy to model different kite types.
- The method should not be dependent on certain computer programs, but be general applicable.

This is an interesting and exciting research goal for several reasons.

- All the existing kite models have their shortcomings that will be solved with this approach: the model will be realistic and fast.
- This will give us more understanding of the flying and steering behaviour of a kite which will lead to better kites designs, controllers and optimal systems.
- It will be a new tool to test new designed kites in a virtual environment before building them.
- It poses an academically interesting challenge. An aero-elastic problem between a highly non-linear flexible kite and an aerodynamic model coupled to a cable model of the bridles and tether should results in a model with low computation times. The approach that will be described in this thesis is a completely new one.

Chapter 4

Approach

This chapter describes a new approach to model a kite so that it fulfils all the in Chapter 3 stated requirements. Section 4-1 discusses the several options to model the structural kite, the aerodynamics and the tether and bridles, followed by a selection of the best options and a full description of the modelling approach in Section 4-2. Section 4-3 describes how this approach will be implemented.

4-1 Modelling options

This section discusses the several modelling options to model a realistic kite in Section 4-1-1, a reduced kite in Section 4-1-1, the tether and bridles in Section 4-1-3 and the aerodynamics in Section 4-1-4.

4-1-1 A realistic kite modelling approach

One of the main requirements states that the model has to represent the real deformations of the kite to realistically capture the global dynamics. Only four of the described modelling techniques in Chapter 2 include the deformations, but not all have the same level of realism.

The multi-plate model of Williams et al. [53] (**option 1**) and the particle model of Furey and Harvey [17] (**option 2**) both introduce some degrees of freedom to include the deformation of the kite in a very rudimentary manner. Due to the coarse discretization, the aerodynamic forces can not be calculated very precisely. The advantage is that these models are fast, because of their limited number of degrees of freedom, but they have two big disadvantages. Firstly they don't include real flexibility of the kite, because all the individual elements are rigid and only rotate with respect to each other. This does not result in the real kite behaviour. Secondly a large number of artificially fitted parameters is needed to mimic the real kite properties. It is very difficult to determine those properties from measurements or calculations, because they are not direct physical properties.

The multi-body approach of Breukels [7] (**option 3**) captures most the global dynamic behaviour of the kite, uses a detailed aerodynamic model and is still relatively fast (20-30 slower than real-time), but has some disadvantages. The multi-body dynamics method is primarily used to couple rigid bodies together with constraints and not to simulate flexible structures. The inflatable beams for instance are simulated with a series of hinged rigid elements with rotational springs to simulate bending and torsion stiffness. This representation does not include the axial strains. The real flexibility of the kite is thus not included. Also many fitted parameters are needed for all the artificially introduced elements and the real physical material properties cannot be used directly.

The finite element model of Schwoll [43] (**option 4**) is based on the finite element method that is widely used to simulate structural flexible problems. The method discretizes a structure into a large number of small finite elements and calculates the strains and stresses in all the elements based on the real material properties. This is currently the method with the highest accuracy for structural problems. The kite model uses a fine mesh with more than 30.000 elements to capture all the dynamic effects in the kite and even the internal pressure of the inflatable beams is modelled in a realistic manner. However, computations can be very slow for structures with many elements. Furthermore there is no aerodynamic model available to produce a realistic aerodynamic pressure distribution on the canopy. The high detail of the finite element model is useless without a correct aerodynamic model, because the deformations will still be wrong.

Because the finite element method is specially designed to model the dynamic behaviour of deformable bodies, it will produce the most realistic model of a kite out of the above mentioned options and therefore fits the *realistic* requirement the best. All the other approaches simplify the physical system too much. Furthermore it is a challenge to create the first flying finite element kite.

4-1-2 A reduced kite modelling approach

The high precision model of Schwoll [43] is too detailed and computational intensive for the new approach. This section discusses four options to reduce a high detailed finite element model to a fast model, without losing the essential global dynamics of the kite.

Option 1: proper orthogonal decomposition with DEIM

Proper orthogonal decomposition (POD) [8, 25, 26, 38, 50] is a technique to reduce large finite element models by finding their most important deformation modes when subjected to their typical loads and use those modes to construct a reduction basis. The deformation of a structure can be rebuilt from these most important deformation modes, reducing the number of degrees of freedom from several thousands to the number of modes that are used in the reconstruction process. The kite is probably well suited for this approach, because its shows a finite number of well defined deformation modes when it is flying. *Jellyfishing* and *twisting* as depicted in Figure 4-1 are the most important deformation modes that can clearly be seen when flying a kite. The dynamic equations of a finite element model can be written in the

linear and non-linear case as (this is explained in more detail in Chapter 5)

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q}) \quad \text{non-linear} \tag{4-1}$$

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{F} \quad \text{linear} \tag{4-2}$$

where $\boldsymbol{f}$ are the internal forces, $\boldsymbol{g}$ the non-linear external forces, $\boldsymbol{F}$ the linear external forces, $\boldsymbol{M}$ the inertia matrix, $\boldsymbol{K}$ the linear stiffness matrix and $\boldsymbol{q}$ the displacements. The displacements $\boldsymbol{q}$ can be approximated from a combination of the most important deformation modes as

$$\boldsymbol{q} = \boldsymbol{\Psi}\boldsymbol{u} \tag{4-3}$$

were the matrix $\boldsymbol{\Psi}$ is an orthonormal basis and contains n vectors with the selected most important deformations modes, called the reduction basis. $\boldsymbol{u}$ are the modal coordinates, the reduced number of degrees of freedom that describe 'how much' of a mode is represented in the current deformation. The most important mode shapes can be found with the POD method from simulation data performed with the full model subjected to its typical loads using singular value decomposition. Combining the above formulas leads to

$$\boldsymbol{M}(\boldsymbol{\Psi}\boldsymbol{u})\boldsymbol{\Psi}\ddot{\boldsymbol{u}} + \boldsymbol{f}(\boldsymbol{\Psi}\boldsymbol{u}) = \boldsymbol{g}(\boldsymbol{\Psi}\boldsymbol{u}, \boldsymbol{\Psi}\dot{\boldsymbol{u}}) + \boldsymbol{R} \quad \text{non-linear} \tag{4-4}$$

$$\boldsymbol{M}\boldsymbol{\Psi}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{\Psi}\boldsymbol{u} = \boldsymbol{F} + \boldsymbol{R} \quad \text{linear} \tag{4-5}$$

where the residual error $\boldsymbol{R}$ comes from the fact that not all deformation modes are used. The projection of this error on the reduction base should be zero. This leads to the new dynamic equations with a greatly reduced number of degrees of freedom

$$\boldsymbol{\Psi}^T\boldsymbol{R} = \boldsymbol{0} \tag{4-6}$$

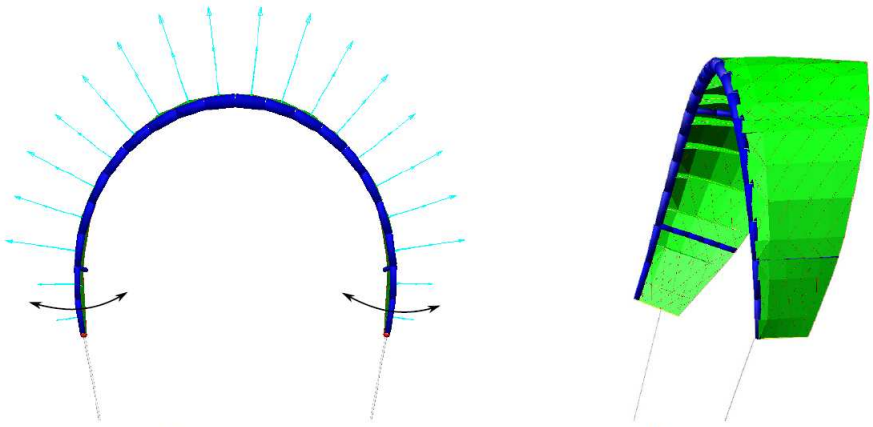
$$\boldsymbol{\Psi}^T\boldsymbol{M}(\boldsymbol{\Psi}\boldsymbol{u})\boldsymbol{\Psi}\ddot{\boldsymbol{u}} + \boldsymbol{\Psi}^T\boldsymbol{f}(\boldsymbol{\Psi}\boldsymbol{u}) - \boldsymbol{\Psi}^T\boldsymbol{g}(\boldsymbol{\Psi}\boldsymbol{u}) = \boldsymbol{0} \quad \text{non-linear} \tag{4-7}$$

$$\boldsymbol{\Psi}^T\boldsymbol{M}\boldsymbol{\Psi}\ddot{\boldsymbol{u}} + \boldsymbol{\Psi}^T\boldsymbol{K}\boldsymbol{\Psi}\boldsymbol{u} - \boldsymbol{\Psi}^T\boldsymbol{F} = \boldsymbol{0} \quad \text{linear} \tag{4-8}$$

Some remarks need to be made. Firstly the equations of the kite are non-linear. Therefore stiffness matrix is not constant but has to be recalculated in every iteration of the Newton-Raphson method as will be explained in Chapter 5 and projected on the newly formed basis. This is a major part of the computations and POD won't reduce this. Krysl et al. [25] shows that POD still can be interesting for the non-linear case when the number of modes grows much more slowly than the number of nodes. Secondly the aerodynamic forces are non-linear and have to be recalculated for every original degree of freedom and then projected on the reduced basis. No speed will be gained here from the POD method. A solution could be to use the Discrete Empirical Interpolation Method (DEIM)[9] to interpolate the external forces. However, this method is very new and has never been used for an application like this before. From a more practical point of view, there is no suitable finite element of the kite that could be used for this reduction method. The model of Schwoll [43] doens't fly because there is no appropriate aerodynamic model and it was developed in commercial software with no option to alter the equations of motion.

Option 2: quasi-static kite model

A flying kite can be seen as a quasi-static process. A quasi-static process happens infinitely slow and can be seen as if it is going through a sequence of equilibrium states. The inertia of a



(a) Jellyfishing **(b)** Asymmetric twist

Figure 4-1: Structural deformation modes of the kite [11].

kite is very small and therefore the inertia forces are very small compared to the aerodynamic forces. This is in fact the case, because the mass of the kite is around 10kg, while the aerodynamic force is around 4000N while flying crosswind. The behaviour of the kite is fully determined by the aerodynamic forces. This makes the assumption valid to neglect the inertia term in the dynamic finite element Equation 4-2 and this results in a static equation that needs to be solved.

$$\boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q}) \tag{4-9}$$

A distinction needs to be made between the local inertia and the global inertia. When only the finite element model of the kite is considered without the tether and bridles, the inertia are called the local inertia that can be neglected as was discussed above. The global inertia of the kite are formed by a kite on a long tether and can be approximated by a point mass on a long tether as mr^2, which is much larger than the local moments of inertia, so those can be omitted in the global simulation as well.

The advantage of treating the kite as a quasi-static structure is that the time-integration algorithm doesn't have to be used for the finite element model which saves a great amount of computation time. The kite can be used as a force generator that exerts forces on a dynamic model of the tether. Time integration only has to be performed on a coupled degrees of freedom in the dynamic tether model.

The exerted forces on the bridle attachment points depend on the degrees of freedom in the dynamic simulation. The fluid-structure-interaction problem doesn't have to be included in the dynamic differential equations if it is assumed that these forces remain constant during a time-step. This means that they don't have to be recalculated multiple times per time-step in a dynamic solving algorithm. This will speed up the simulation with a factor 4 (depending on the solver). This will be discussed in more detail in Chapter 8.

Option 3: black box approach

A full finite element model could be reduced by performing many simulations for various flight conditions and trying to find a mapping between the inputs of the model (wind speed,

steering input, etc.) and the outputs (forces on the bridle lines). This could be formulated as a look-up table where the forces can be found that correspond to a certain state. This black-box model could then be coupled to a dynamic simulation of the tether. This is not a very flexible approach, because the whole procedure would have to be redone if the kite model changes, but it will result in a very fast model.

Option 4: simplify the finite element mesh

The model complexity can be reduced by using a coarse mesh, because only the global dynamics are of interest and not the local dynamic effects or wrinkling of the canopy and inflatable beams. Coarsening the mesh of the canopy is quite straightforward.

The inflatable beams can be modelled with regular beam elements that mimic the material properties of inflatable beams, which will result in the same dynamic behaviour. Inflatable beams have to be meshed with many small shell elements on the surface of the beam, while a section of a normal beam can be modelled with just one element. This greatly reduces the amount of needed elements to represent the beams and therefore speeds up the calculations.

Furthermore some simplifications of the kite can be made on parts that are not important to model into detail, for example the connection between different parts of the kite.

4-1-3 Tether and bridles

This section discusses the options to include the tether and bridles in the simulation. The method should be able to model the bridles of different kite types as discussed in Section 2-1-2.

Option 1: Include the tether and bridles in the finite element model

The bridles and tether can also be modelled in the finite element model as depicted left in Figure 4-2. This can either be done by modelling them as simple spring elements or as geometric multipoint constraints [2, 35, 24] that describe the relations between the by the bridles constrained nodes in the finite element model. Constraints also have to be used together with the spring elements when pulleys are used to connect multiple bridles. Some issues arise in both approaches.

- The multipoint constraints have to be included in the dynamic equations. The stiffness matrix will lose its symmetric properties and thereby some calculation advantages.
- The kite can not be used as a quasi-static force generator, but has to be modelled fully dynamic.
- The resulting model has rigid body modes that introduce large rotations in the model. The co-rotational framework has to be used in the element definition to correctly calculate the stresses and handle the rigid body modes. This method decouples the deformation modes from the rigid body modes by using different reference frames. This would be an interesting approach, but not very practical for reasons that are discussed Chapter 5.

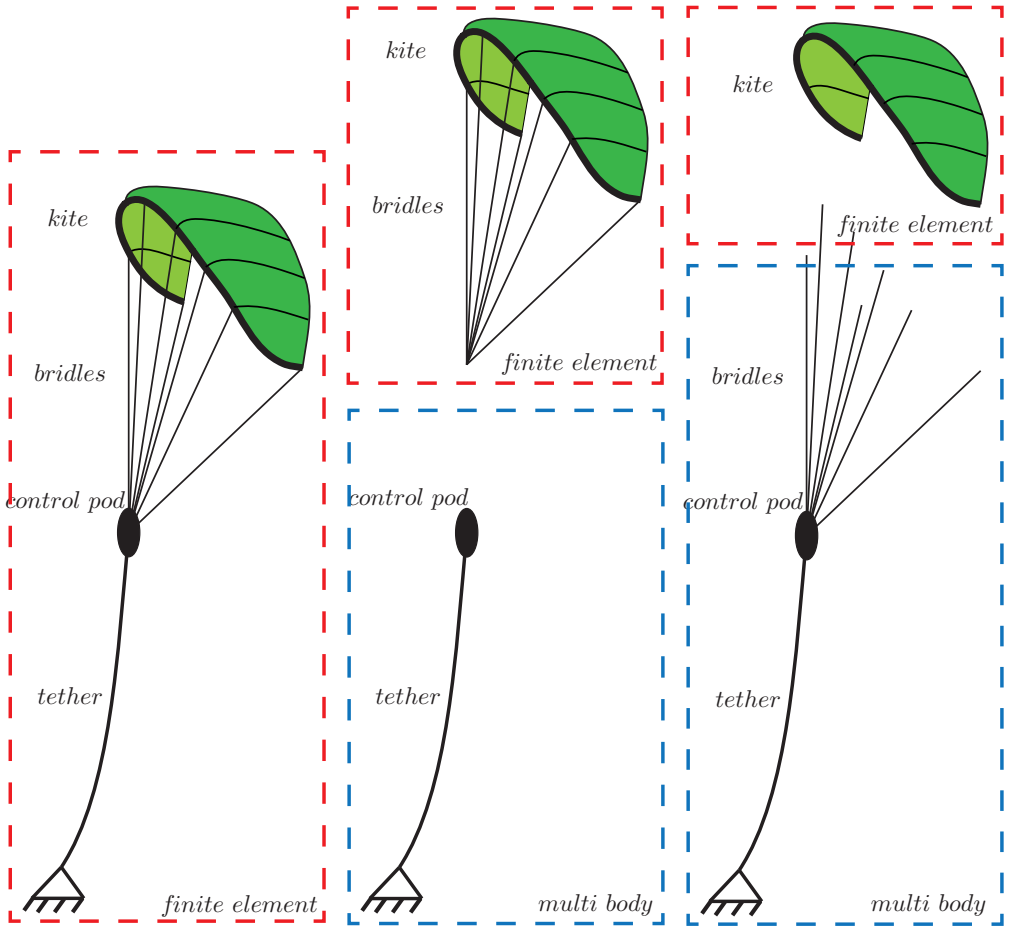


Figure 4-2: Three options to model the system. Model everything with finite elements (left), model the kite with the bridles with finite elements and the tether with the pod with a dynamic approach (middle), model the kite with finite elements and model the bridles, pod and tether with a dynamic approach(right).

- It is difficult to use POD and DEIM in this approach.

Option 2: A finite element kite and bridle model with a dynamic tether

The tether and control pod could be modelled with a dynamic approach as discussed in Chapter 7 and connected to a finite element model of the kite with the bridles as depicted in the middle of Figure 4-2.

The coupling can be done in two ways. The kite with bridles can be used as a quasi-static force generator that exerts forces on the control pod. The dynamic equations of the finite element system can also be coupled to the dynamic equations of the tether and bridles system, forming one large system of dynamic equations [51, 44]. This will result in asymmetric matrices and will lose calculation advantages in the finite element equations.

Similar problems as in option 1 arise. Rotational rigid body modes are still present in the model, although the largest rotations are now handled in the dynamic tether model which

makes a co-rotational framework less important. The problem with another framework is that the rigid body modes have to be eliminated from the equations, which is not straightforward in non-linear analysis, because the stiffness matrix changes in every iteration.

Option 3: A finite element kite model with dynamic bridles and tether

The bridles can be modelled together with the tether in the dynamic simulation as depicted on the right in Figure 4-2. The rigid body modes and the deformation modes are now split, because the kite is fully constrained in its reference frame at the bridle points, removing the necessity to use a co-rotational framework. The finite element model only has to deal with the deformations, while the dynamic model handles the rigid body modes. A floating reference frame is attached to the end of the bridles that rotates with the dynamic simulation as depicted in Figure 4-3 and forms the base reference frame for the finite element model. This method of decoupling rigid body modes and deformation modes was mentioned before by B.Fraeijs and Veubeke [6] as the shadowing problem, because the floating frame follows the structure as a shadow.

The coupling can again be done in two manners. Both models can be coupled together producing one large dynamic system of equations as was explained in option 2. But the finite element kite model can also be used as quasi-static force generator that exerts forces on the end points of the bridles. Both models are coupled together by forces and displacements. The dynamic model of the bridles calculates new displacements that are used as constraints in the finite element calculations. The finite element model calculates the resulting forces on he bridle attachment points and returns them to the dynamic bridles simulation. The forces need to act on a mass and therefore the kite mass has to be distributed over the bridle attachment points. The exact distribution of the mass over the bridle points is less important, because the main dynamics are determined by the aerodynamic forces. This can be compared to a spring-damper system with a very high damping constant and a very small mass (drag forces and the mass of the kite), where the total dynamics of the system are mostly determined by the damping coefficient and not by the size of the mass.

4-1-4 Aerodynamic modelling approach

It is important to use an aerodynamic model that matches the level of detail of the structural model, because the deformation of the kite fully depends on the aerodynamic forces. Modelling the aerodynamics is very complicated and not the focus of this thesis, but an appropriate model needs to be selected to show the modelling approach. This model can always be replaced by other models in the future.

The main criteria are that the model should be reasonably fast because the forces need to be recalculated in every iteration step, easy to implement and give a realistic force distribution that depends on the deformation of the kite.

Out of the available models listed in Chapter 2 only CFD and the Breukels [7] model return a distributed load that depends on the deformation of the kite. Because CFD is much too slow, the Breukels [7] model is the only real option. The model was verified to some extent and produced results that were accurate enough to use in his kite modelling approach. It is fast and easy to implement, since the description is available.

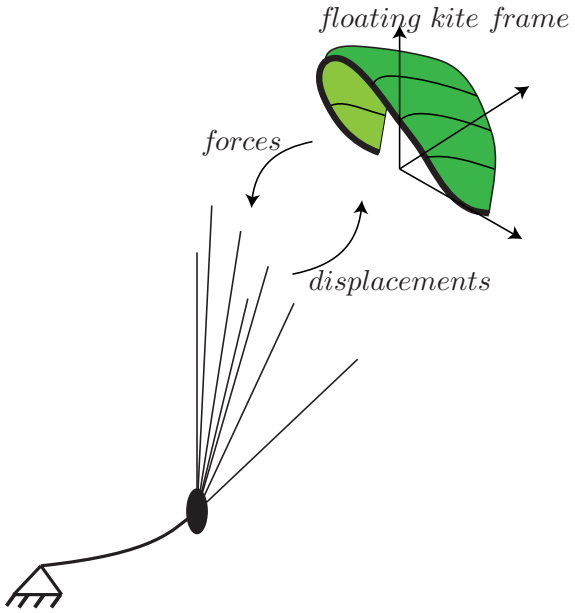


Figure 4-3: A finite element kite in a floating reference frame that is attached to the end points of the bridles in a dynamic simulation. The dynamic simulation of the bridles gives displacements information to the finite element model that returns forces to the dynamic model.

4-2 The full model description

A combination of options from the previous sections was made that results in an approach that fulfills all the requirements from Chapter 3 the best.

The kite will be modelled with the finite element method (option 4) and coupled to the Breukels [7] aerodynamic model to obtain the most **realistic** approach, because the finite element model is based on physical properties of the system and fully incorporates the flexibility of the kite.

This model will be **reduced** to a quasi-static model (option 2) with a simplified mesh (option 4) and coupled to a dynamic model of the tether and bridles (option 3) with the assumption that the forces on the bridles remain constant within a time-step. This greatly increases the computational speed, because the dynamic integration only has to be done for the degrees of freedom in the dynamic model and the fluid-structure-interaction problem only has to be solve once for each time-step, without losing the important global dynamics of the kite.

Figure 4-4 shows this approach schematically. The dynamic integration is done in the tether and bridles model. This model calculates in every time-step the displacements of the bridle attachment points in a floating reference frame that is the base reference frame for the finite element model. The quasi-static fluid-structure-interaction problem between the finite element kite and the aerodynamic model is solved and returns forces to the dynamic tether and bridles model. Those forces are exerted on the end points of the bridles.

This approach has several more advantages.

- The finite elements can be modelled with the Total Lagrangian framework.

- The finite element stiffness matrix remains symmetric.
- The tether and aerodynamic model can easily be changed.
- This model can even be further reduced with POD/DEIM or the blackbox approach.
- It is a flexible method that makes it easy to model a different bridle system or a kite.

The next chapters go into detail on the different parts of this approach. Chapter 5 sets up the finite element model, followed by the aerodynamic model in Chapter 6. The dynamic tether and bridles model will be discussed in Chapter 7. All components will be integrated in one system in Chapter 8.

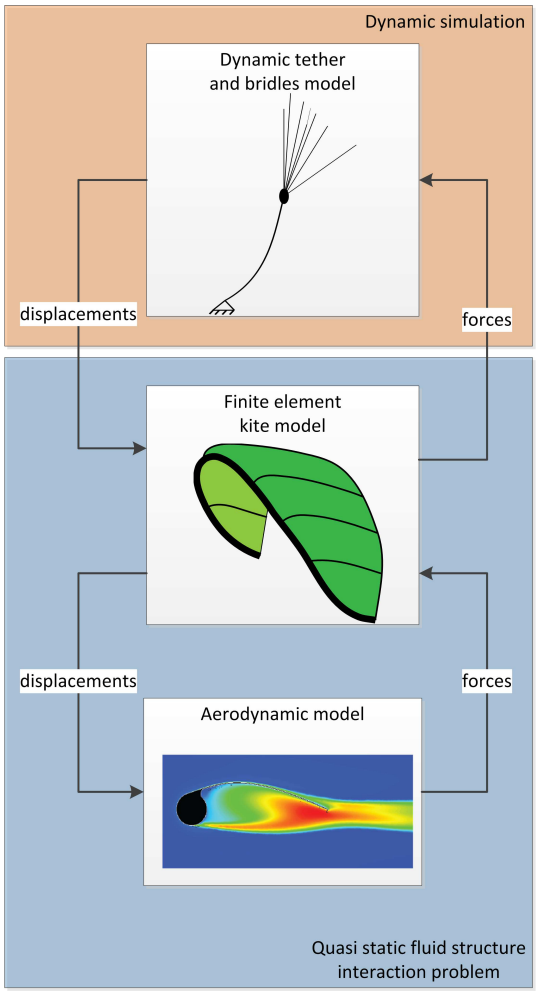


Figure 4-4: A schematic of the new modelling approach. A quasi-static finite element kite model and an aerodynamic model form a fluid-structure-interaction problem together that is coupled to a dynamic simulation of the tether and bridles.

4-3 Implementation

The main criteria for the implementation are listed below.

- Speed: the result should be fast.
- Flexibility: it should be easy to alter the code and try new things.
- Independence: the model should run on every computer and not rely on certain programs.
- Interface: the model should be able to communicate easily with developed controllers, tether models and ground station models.

Commercial finite element programs like Ansys, Abacus or Madymo can be used to build the finite element model. These programs can interface with other programs that do the dynamic tether and bridles calculations. The advantage of commercial software is that many element types are available, that the calculations are optimized for speed and that the models can easily be built in the graphical interface. The disadvantage is that the flexibility is low. There is no option to alter the dynamic equations and try to implement new reduction algorithms. Furthermore the model will only run on computers with the software installed. Open source finite element programs like CalculiX or Code Aster can be used as well. These give a bit more flexibility to alter the code, but are still not the ideal environments to play around with new simulation techniques, because it involves profound knowledge of the program.

All the components could also be implemented in MATLAB , giving more flexibility. The finite element implementation will be slower than in commercial software, because the MATLAB language is not compiled. It has the option to compile parts of the code to (non-optimized) C++, which will speed up the model a bit. This approach will involve much more work, because the full finite element model has to be programmed. Fortunately parts of the code from Tiso [47] can be used for this.

If the complete code can be written in MATLAB (and compiled) it can work as a stand-alone and can easily be coupled to different controllers that were or instance developed in Simulink or C++.

Because flexibility is more important than speed when developing a new method, MATLAB will be used as the platform to test this new approach. If the approach turns out to be successful, it can of course be implemented in different codes, on different platforms or be used together with commercial finite element software.

Chapter 5

Finite element kite model

This chapter describes the modelling of the kite with the finite element method. Section 5-1 briefly introduces the fundamentals of the finite element method, derives the governing equations and discusses how the non-linear behaviour of the kite is included in the model. Section 5-2 describes the modelling of the different parts of the kite: the canopy, the leading edge and struts, the trailing edge wire, the tips, the external forces and the boundary conditions. The solving algorithm is discussed in Section 5-3. The chapter concludes with some final remarks in Section 5-4.

5-1 The non-linear finite element method

This section introduces the basic concept of the finite element method and the important non-linear behaviour that the kite shows and is included in the dynamic equations.

5-1-1 Introduction to the finite element method

The finite element method is a computational technique to solve complex partial differential equations by discretizing the domain into a finite number of finite elements. Each element contains points at which the solution to the differential equations is calculated explicitly, called the nodes. The unknowns are the field variables at the nodes, in the case of structural problems as the kite the displacements. The values of the displacements can be approximated everywhere in the element via the interpolation with the shape functions. Consider for instance the two dimensional field as depicted in Figure 5-1 with the displacement in two directions as field variables. The field is discretized with triangular elements with three nodes. The displacement field of an element $\boldsymbol{u}_e(\boldsymbol{x},t)$ at time t can be approximated everywhere in an element with the shape functions $\boldsymbol{N}_e(\boldsymbol{x}) = [N_1(\boldsymbol{x}) \quad N_2(\boldsymbol{x}) \quad N_3(\boldsymbol{x}) \quad N_4(\boldsymbol{x}) \quad N_5(\boldsymbol{x}) \quad N_6(\boldsymbol{x})]$ and the nodal displacements of an element $\boldsymbol{q}_e(t) = [u_1 \quad v_1 \quad u_2 \quad v_2 \quad u_3 \quad v_3]^T$

$$\boldsymbol{u}_e(\boldsymbol{x},t) = \boldsymbol{N}_e(\boldsymbol{x})\boldsymbol{q}_e(t) \tag{5-1}$$

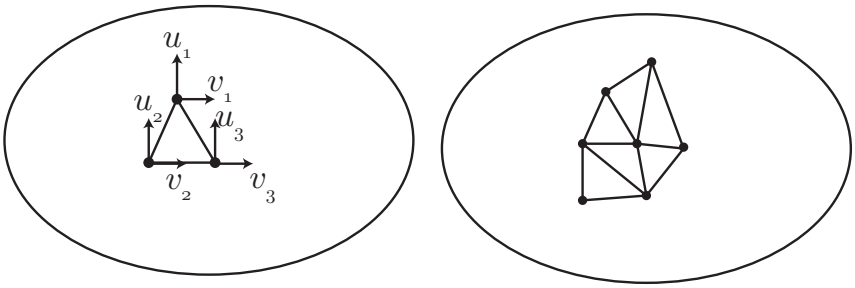


Figure 5-1: A field is discretized with triangular finite elements. The displacements u and v are the field variable. Every element has six degrees of freedom $\boldsymbol{q}_e$.

All the degrees of freedom are denoted by $\boldsymbol{q}$. The accuracy of the obtained solution strongly depends on the amount of used finite elements and the type of shape functions. A very large number of different element types exist with different shapes, number of nodes, shape functions and based on different assumptions, all suitable for different situations. The finite element method is very powerful for calculations on complex systems. Unfortunately it can also be a very computational expensive method for large systems with fine meshes. More on the finite element method can be found in introductory books, like Hutton [22].

5-1-2 Equations of motion

The equations of motion inside a continuous body are described by the partial differential equations [40]

$$\frac{\partial}{\partial x_i}(\sigma_{ij} + \sigma_{im}u_{j,m}) + X_j - \rho\ddot{u}_j = 0 \qquad j = 1, 2, 3 \tag{5-2}$$

and express the equilibrium of a deformed infinitesimal volume in the body. X_j are the components of the body forces (like gravity), ρ is the mass density of the material, σ are the stresses, x_i are the three dimensional coordinates and u are the displacements in three directions. One method to derive the discretized equations in finite elements is by writing the potential and kinetic energies of the system in terms of the shape functions and nodal displacements and using variational calculus to obtain the equations of motion.

The potential energy for an element with linear elastic material is expressed by

$$\mathcal{P}(\boldsymbol{q}_e) = \frac{1}{2}\int_{V^e} \boldsymbol{\varepsilon}^T(\boldsymbol{q}_e)\boldsymbol{H}\boldsymbol{\varepsilon}(\boldsymbol{q}_e)d\mathrm{V} \tag{5-3}$$

where $\boldsymbol{\varepsilon}(\boldsymbol{q}_e)$ are the stresses and $\boldsymbol{H}$ the Hooke matrix containing the linear elastic coefficients. The strains are in general non-linear equations of the displacements.

The internal forces on the nodes of an element can be written as

$$\boldsymbol{f}_e(\boldsymbol{q}_e) = \frac{\partial \mathcal{P}}{\partial \boldsymbol{q}_e} \tag{5-4}$$

The tangential stiffness matrix $\boldsymbol{K}_t$ results from the second derivative of the potential energy and depends in general on $\boldsymbol{q}_e$

$$\boldsymbol{K}_e(\boldsymbol{q}_e) = \frac{\partial^2 \mathcal{P}}{\partial \boldsymbol{q}_e \partial \boldsymbol{q}_e} \tag{5-5}$$

When small deformations are assumed $\boldsymbol{\varepsilon}(\boldsymbol{q}_e)$ becomes a linear function of $\boldsymbol{q}_e$ and the stiffness matrix becomes constant.

The mass matrix $\boldsymbol{M}_e$ for an element is found from the kinetic energy expression

$$\mathcal{T} = \frac{1}{2}\int_{V_e} \rho \boldsymbol{u}_e^T \boldsymbol{u}_e d\mathrm{V} = \frac{1}{2}\boldsymbol{q}_e^T \int_{V_e} \rho \boldsymbol{N}_e^T \boldsymbol{N}_e d\mathrm{V} \boldsymbol{q}_e = \frac{1}{2}\boldsymbol{q}_e^T \boldsymbol{M}_e \boldsymbol{q}_e \tag{5-6}$$

and can be written as

$$\boldsymbol{M}_e = \int_{V_e} \rho \boldsymbol{N}_e^T \boldsymbol{N}_e d\mathrm{V} \tag{5-7}$$

This leads after assembling all the contributions from the elements to the inertia forces and internal forces to the discretized equations of motion

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{g}(t) \tag{5-8}$$

where $\boldsymbol{M}$ is the assembled mass matrix, $\boldsymbol{f}$ the assembled internal forces and $\boldsymbol{g}(t)$ the nodal external forces. In the linear case where the internal forces don't depend on $\boldsymbol{q}$, the equations can be written as

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{g}(t) \tag{5-9}$$

The direct stiffness method will be used to assemble the mass matrix, stiffness matrix and internal force vector and is described in all introductory books on the finite element method.

The hypothesis in this thesis is that the inertia of the kite can be neglected in the local finite element calculations due to the quasi-static assumption of the kite as explained in Chapter 4, leaving a static problem. Furthermore the external forces, mainly coming from the aerodynamic forces, also depend on the displacements $\boldsymbol{q}$ and the overall state of the system $\boldsymbol{X}$. $\boldsymbol{X}$ contains the kite velocity $\boldsymbol{V}_k$, the rotational kite velocity $\boldsymbol{\omega}_k$, the velocity of the nodes $\boldsymbol{V}_{struct}$, and the wind velocity $\boldsymbol{V}_w$. The external forces are explained in more detail in Section 5-2-8. The initial configuration $\boldsymbol{q}_0$ is prescribed, either by the solution from the previous time step in the dynamic tether and bridles simulation or set to $\boldsymbol{0}$ when a single case is solved. The boundary conditions $\boldsymbol{q}_b$ contain the prescribed degrees of freedom at the bridle attachment points, also resulting from the dynamic simulation. This altogether results in the following static non-linear equations

$$\begin{cases} \boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{X}) \\ \boldsymbol{q}_0, \boldsymbol{q}_b, \boldsymbol{X} \qquad \textit{prescribed} \end{cases} \tag{5-10}$$

5-1-3 Non-linearities in finite elements

Most structural systems can be approximated with linear systems when the deformations remain small. Unfortunately this isn't the case for the kite and therefore the non-linear equations have to be solved, which is much more computational intensive, because they have to be solved in an iterative manner and the stiffness matrix needs to be recalculated in every step.

Non-linearities can be recognized from a load-deflection response diagram as showed in Figure 5-2 and explained by Felippa [14]. It shows the equilibrium path of a structure that

deforms under an applied load. Every point on the path represents a possible static equilibrium configuration of the structure. If this path is non-linear it means that the system is non-linear. Special points can arise in the path, for instance critical points at which more

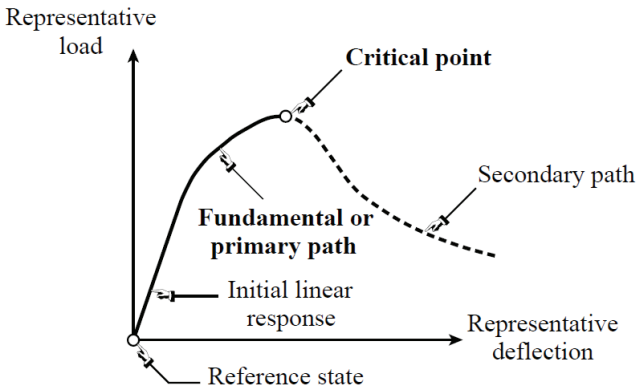


Figure 5-2: The load-deflection response diagram. Felippa [14].

equilibrium paths cross each other or when the tangent to the equilibrium path becomes zero. Responses can get really complicated: softening, hardening, snap-through, snap-back and buckling (bifurcation). The latter one happens often in thin structures, as is the kite. At bifurcation points more response paths are possible, which makes it a difficult problem to solve.

A linear system doesn't show this behaviour. It can withstand any load and undergo any displacement which is never realistic, but can be a good approximation around certain equilibrium configurations. However, when the configuration diverts to far from this configuration it is necessary to look into the non-linearities.

The stiffness of a system is the tangential stiffness around a certain point of the equilibrium path. In the linear case this is a constant, in the non-linear analysis the tangential stiffness matrix changes along the equilibrium curve and therefore depends on the current configuration $\boldsymbol{K}_t(\boldsymbol{q})$.

In structural analysis there are four sources for non-linear behaviour. This can nicely be seen from Figure 5-3. The figure describes the relations between the different fields in solid continuum mechanics. In every one of these relations non-linearities can arise.

- Geometric (kinematic equations and equilibrium equations): the change in geometry is taken into account when setting up the displacement and equilibrium equations.
- Material (constitutive equations): the material behaviour is non-linear and depends on the current deformation, deformation history or parameters such as temperature, pre-stress or moisture.
- Force boundary conditions: the applied forces depend on the deformation.
- Displacement boundary conditions: the applied displacements depend on the deformation of the structure.

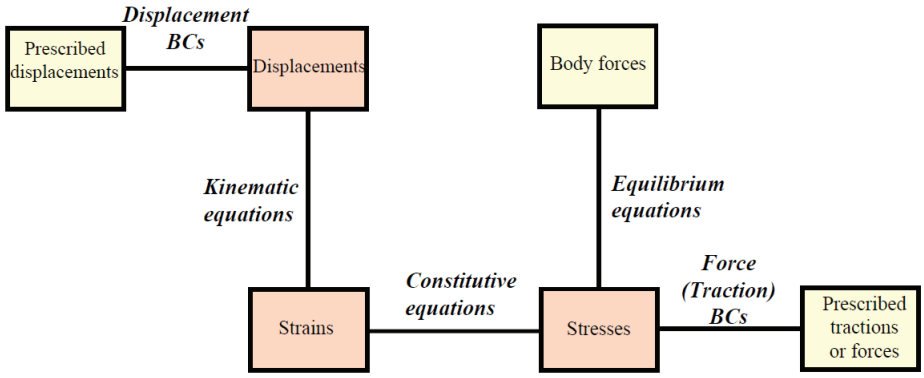


Figure 5-3: Relations between the fields in solid continuum mechanics Felippa [14].

A non-linear system has to be solved in a iterative manner, because the stiffness of the system depends on its displacements and the other way around. During this iteration process the stiffness matrix needs to be recalculated in every step, making solving a non-linear problem is much more expensive than solving a linear system. More details about solving a non-linear system are given in section 5-3.

5-1-4 Non-linearities in the kite

Three of the four mentioned sources for non-linear behaviour are present in the kite.

Geometric non-linearities are an important source of non-linearities in the kite. The deformations of the kite are large and the stiffness changes completely when the kite changes to a different configuration. This can clearly be seen from the canopy. The canopy is very thin and has almost no bending stiffness in the direction of the applied out-of-plane aerodynamic forces when it is in the undeformed state. In the linear case, this would result in very large displacements of the canopy, because there is almost no stiffness in that direction. This is of course not what happens. Because the canopy starts to bend, the direction of the stiffness in the system changes and the canopy will take the load with its axial stiffness and therefore not displace as much. This process is illustrated in Figure 5-4. These non-linearities come from the kinematic description of the strains ε that depend non-linear on the displacements $\boldsymbol{q}$ The difference between a linear and non-linear geometric analysis of the kite subjected to the same load is shown in Figure 5-5.

Material non-linearities come from the inflatable leading edge and struts. The beams will be modelled with regular beam elements as will be explained in Section 5-2-5. The material properties of the beams become non-linear to give the beams the same behaviour as the inflatable beams. The Young's modulus changes depending on the deformation of the beam and therefore the constitutive equations that describe the relations between stress and strain are not linear anymore.

Force non-linearities come from the aerodynamic forces. The forces depend fully on the kite displacements $\boldsymbol{q}$ and form a fluid-structure-interaction problem together with the kite.

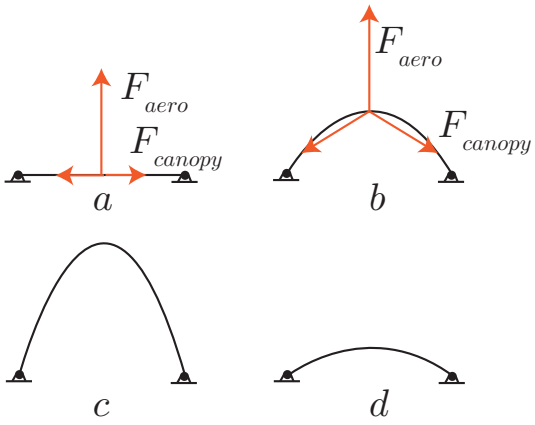


Figure 5-4: Forces an the canopy due to an aerodynamic load. a: undeformed, b: deformed, c: linear response, d: non-linear response.

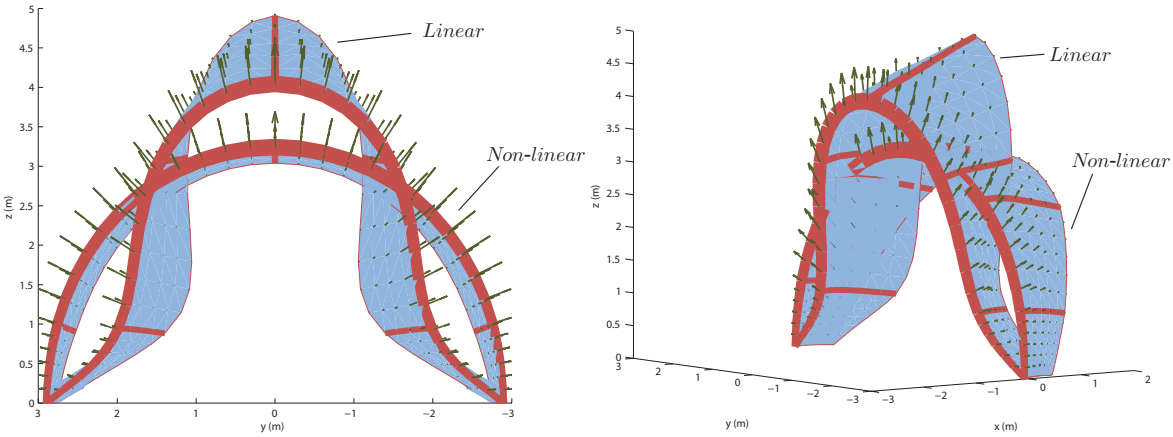


Figure 5-5: A geometrical linear and non-linear solution of the finite element kite model, both subjected to the same aerodynamic loads. No scale factor was applied.

Buckling (bifurcation) is another non-linear effect seen in kites. Figure 5-6 shows an example where the leading edge starts to buckle. These effects occur when the aerodynamic forces get too high. Buckling of the canopy is a problem that will occur more often during the simulations. During some flying conditions the canopy will partly be subjected to compressive loads that will cause the canopy to buckle. This does not happen very often, because the aerodynamic forces are most of the time directed in the outward direction. But sometimes the tips get a negative angle of attack, inverting the directions of the aerodynamic forces, resulting in buckling. The canopy can also start to fold in lines when the kite makes a sharp corner. Stalling the kite will also result in buckling of the canopy. Figure 5-7 shows some examples of this behaviour. All these buckling situations are tried to avoid, because they will result in convergence problems.

These considerations show the necessity to do a non-linear finite element analysis of the kite.

Figure 5-6: Buckling of a kite under high loading.

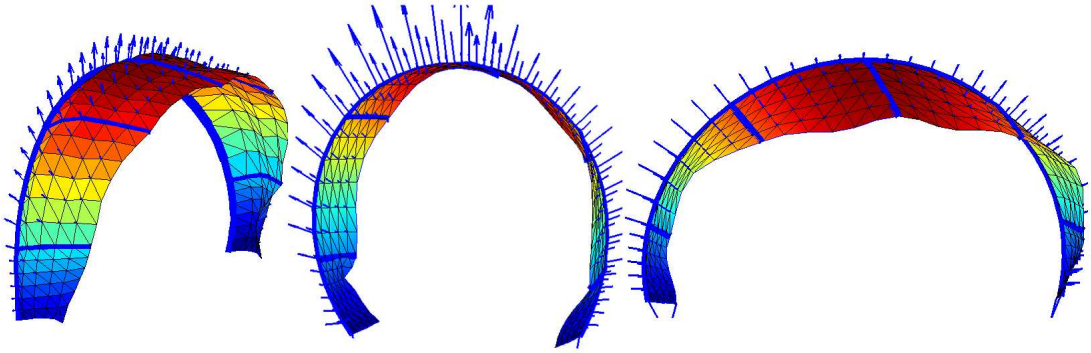
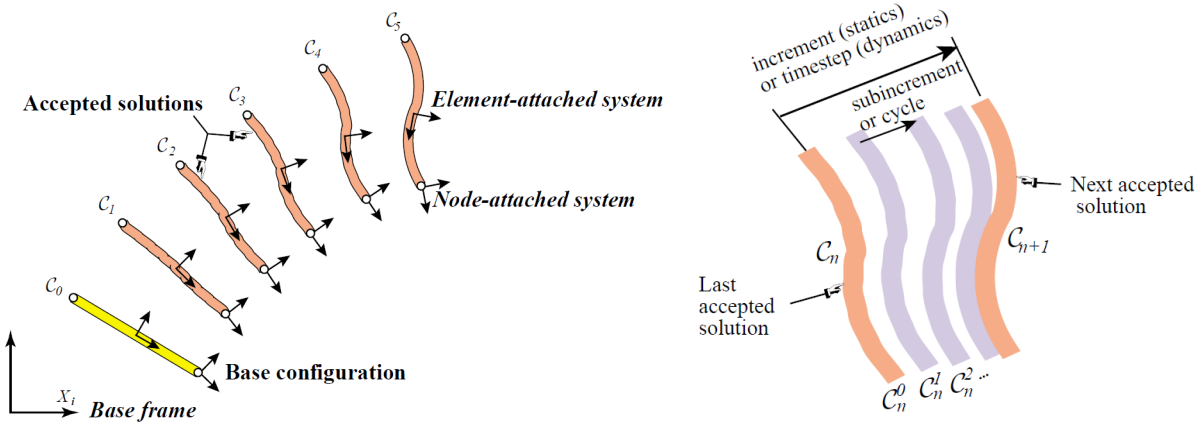


Figure 5-7: Some examples of buckling in the canopy in different flying conditions.

5-1-5 Mathematical framework for element definitions

Finite elements can be developed in different frameworks as described by Felippa [15]: the Total Lagrangian, the Updated Lagrangian or the Corotational framework. Each framework uses a different reference configuration to calculate its variables in. The finite element as depicted in Figure 5-8a is undergoing a series of changes in load. The element starts from an initial configuration C_0 and solving the finite element equations for the different loads results in new configurations C_n after stage n. The system drags two coordinate systems along: the element attached system moves with the element and the node attached systems move with the nodes. There is also a base frame that is fixed and is used to track the motion of the elements.
In non-linear finite elements calculations there are sub-stages which the system undergoes to reach new configuration, as shown in Figure 5-8b. To check whether one of the sub-stages is an acceptable final solution it must be verified if it satisfies the governing finite element equations to a certain accuracy. This needs to be done by checking the state (forces, displacements, strains, stresses, etc.) of the new configuration with respect to a certain reference configuration. Here a choice between the frameworks needs to be made.

1. The Total Langrangian (TL) framework: the FEM equations are formulated with respect to a fixed reference configuration that doesn't change throughout the analysis.



(a) A finite element going through a sequence of converged configurations in a stepwise manner.Felippa [15]

(b) Going from one solution to a new one, undergoing sub-stages.Felippa [15]

Figure 5-8: The solution process of a finite element.

2. The Updated Langrangian (UL) framework: the FEM equations are formulated with respect to the last calculated accepted configuration.

3. The Corotational (CR) framework: the FEM equations make use of two systems, a fixed reference configuration C_B as in TL and a corotated reference configuration that goes along with the element and describes the rigid body motion of C_B.

All the methods have their (dis)advantages. UL is mainly used for very particular applications such as metal forming and therefore not interesting for this project. It is more interesting to look at the (dis)advantages of the CR framework compared to the TL framework with respect to the modelling of a kite.

Advantages of CR compared to TL:

1. Problems with large rotations can be handled, because the stress calculation are still correct for large rotations. In the TL model, the calculated stresses become invalid when the rotations become too large. That the CR description can handle this, is very advantageous for aerospace structures, thus also for a kite.

2. Decouples the material non-linearities from geometric non-linearities, which are both present in the kite.

3. Handles anisotropic behaviour of materials without having to recalculate all the strain directions. Ripstop material of the canopy shows this type op behaviour, although not modelled in that way in this research.

4. Interfaces naturally with multi-body dynamics programs. So coupling with a dynamic simulations of the tether and bridles would be easier.

5. Is well suited for the use of structural elements with rotational degrees of freedom.

Disadvantages of CR compared to TL:

1. Only small strains can be handled.
2. Can lead to unsymmetrical tangent stiffness matrices.
3. Involves difficult mathematics.

It would be nice to develop the whole model in the CR framework, because that would make it possible to model the bridles and tether also with the finite element method. Large rotations and displacements are no problem, because the framework decouples rigid body modes and deformations. However, no suitable element types are available for the CR framework in MATLAB to model the canopy and developing them is not in the scope of this research. A shell element description in the TL framework is available in Matlab from previous research done by Tiso [47] which could be used. Furthermore there is no advantage in the CR framework when using the proposed approach to combine a dynamic tether and bridles model with the quasi-static finite element kite model, since the rotations of the kite will remain small in the floating kite reference frame. Therefore it is decided to work in the TL framework. Although for future work it might be interesting to look into developing a kite in the CR framework.

5-2 Kite modelling

This section discusses how the several parts of the kite are modelled with finite elements.

5-2-1 Modelling requirements

The main requirements for the modelling are stated as follows. The model should

- be able to capture the important global deformation modes and show the real flexibility of the kite;
- use elements that are able to handle the geometric and material non-linearities;
- be able to handle large deformations.
- be as fast as possible without losing essential information;
- solve the fluid-structure-interaction problem between the aerodynamic forces and the kite.

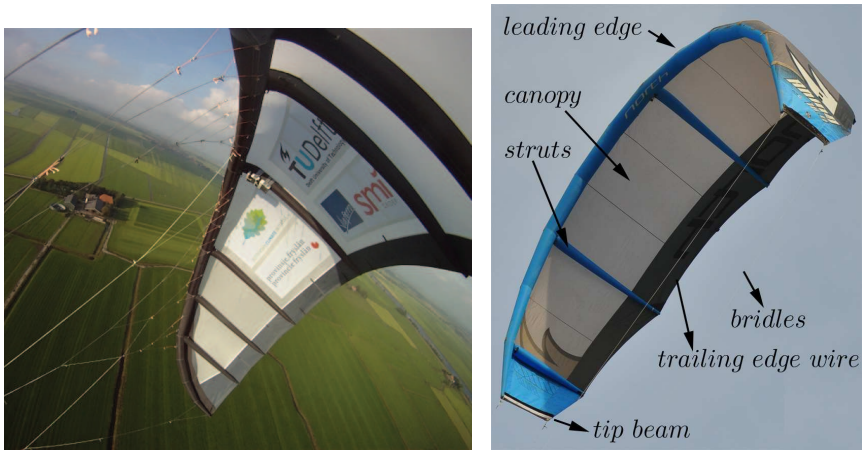
5-2-2 Kite selection

A large variety of kites exists on the market nowadays as was illustrated in Figure 2-7. The Kite Power group from the Delft University of Technoloy has tested several of these kites for power generation and is developing new kites specially designed for power generation. Currently the $25m^2$ Mutiny kite as depicted in Figure 5-9a is used. It has a complex bridle

system of 24 bridles with pulleys to withstand higher forces without collapsing. A different kite will be used in this thesis: the North Rhino $16m^2$ as depicted in Figure 5-9b. This is a kite with four bridles and a single curved leading edge. The kite will be used for the following reasons.

- Since the main goal of this thesis is to propose a new modelling approach, it is more straightforward to consider a kite that contains all the essential features and avoids unnecessary complicated parts as a complex bridle system with pulleys and a double curved leading edge of the Mutiny kite.
- The aerodynamic model from Breukels [7] that will be used in this research was designed and tested for this kite. The model has some limitations that make it more difficult to use it for kites with different shapes as will be discussed in Chapter 6.

It will be shown that the proposed modelling process can also handle more complex kites.



(a) The $25m^2$ Mutiny kite.

(b) The North Rhino $16m2$ kite.

Figure 5-9: Two different kite types.

The North Rhino is a C-shaped kite with an inflatable leading edge and five inflatable struts. The tips contain a thin aluminium beam and four attachment points for the bridles. The inflatable leading edge consist of an outer shell that is made from the material Dacron an in inside bladder (balloon) that is airtight. The trailing edge contains a thin wire to prevent over-stretching of the trailing edge and flapping. The canopy is stitched to the top of the leading edge and the struts and made from thin ripstop material. Appendix C shows some pictures with details of the kite.

5-2-3 Geometric dimensioning & meshing

Geometric dimensioning

The global dimensions of the undeformed and unloaded kite are shown in Figure 5-10. Dimensioning the kite is not straightforward for several reasons. The kite has been built from

drawings, but the manufacturing is handmade and never the same as the drawings. Furthermore the shape of the kite deforms when the leading edge and the struts are filled with air, depending on the pressure. It is also difficult to measure the kite in an unloaded state, since it is so flexible and therefore always deformed by its own weight. The used dimensions were based on available data from de Groot [11], but were altered on some points to produce a more realistic kite. The leading edge and trailing edge of the kite are described by half of an ellipse,

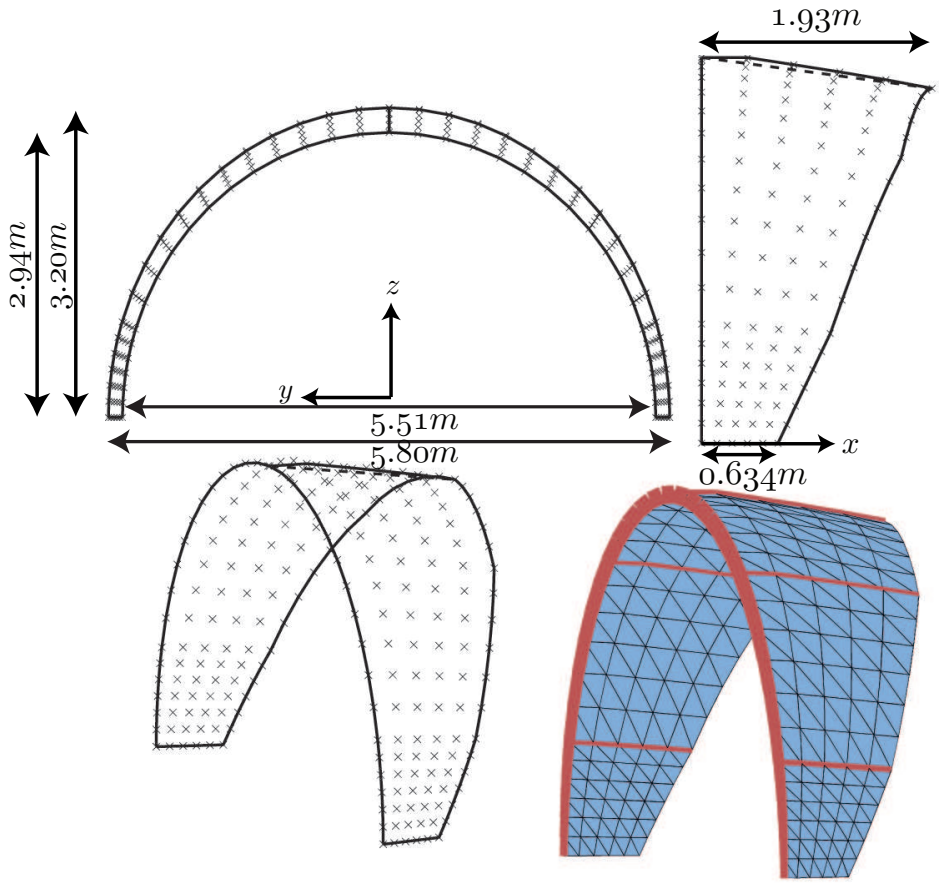
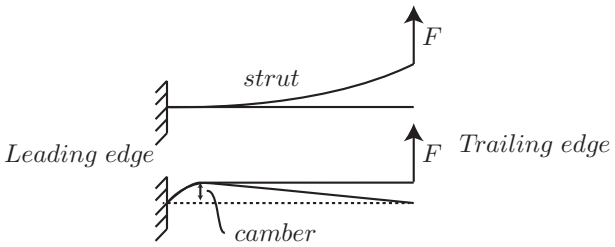


Figure 5-10: The mesh of the North Rhino $16m^2$ kite.

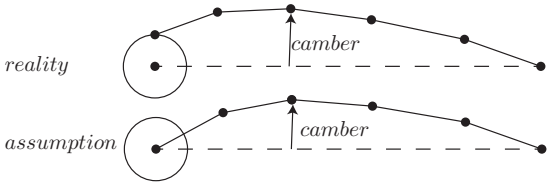
but the span and height of the trailing edge are lower than the leading edge. This creates two effects that are necessary for a stable, good flying kite, but this effect was not included by the original design of Breukels [7] and de Groot [11]. Firstly it creates an initial angle of attack at the tips and the middle section. Secondly it prevents the kite from bending too much forward as was seen from simulations with the kite without this effect included, because the trailing edge is shorter than the leading edge. The parameters were chosen to produce a kite with an initial angle of attack at the tips of 6.5 deg and 12.7 deg in the middle. Kite designer use similar values for their designs and after experimenting with different parameters, this turned out to produce the most stable kite.

Something else that was not in the design of Breukels [7] and de Groot [11] was an initial camber in canopy and struts. Normally struts get an initial camber for better aerodynamic performance. This also influences the structural performance of the kite. If the struts are not

pre-bended they will bend like a clamped beam, resulting in a negative camber. Giving the struts an initial camber will result in better bending behaviour as can be seen in Figure 5-11a. The canopy that is attached to the struts gets the same initial camber. More details on the



(a) Difference between bending of a strut without (top) and with (bottom) an initial camber.



(b) Attachment of the canopy to the midpoint instead of the top of the beam results in a underestimation of the camber.

Figure 5-11: Modelling the initial camber.

dimensions of the kite can be found in Appendix C.

Meshing

The in Chapter 2 mentioned finite element kite model of Schwoll [43] contains over 30.000 elements and models the air flows in the struts and leading edge. This makes computations very demanding and results in long cpu times. Since the goal of this thesis is to propose a new reduced approach that only captures the global dynamics and deformations, some simplification are made with respect to the finite element modelling.

Regular beam elements will be used to model the inflatable leading edge and struts with similar material properties as was discussed in Chapter 4. The trailing edge wire and rods in the tips will also be modelled with these beam elements. Shell elements will be used to model the canopy. Details of the elements will be discussed in the next sections.

Simplifications were made in the meshing process.

- The canopy is attached to the same nodes as the struts and leading edge. This means that the canopy is physically attached to the middle of the beams and not on top of them. This will probably result in an underestimation of the camber as shown in Figure 5-11b.
- The aluminium tip bar, the leading edge and the trailing edge wire are for all connected to each other by the corner nodes. In reality this is not the case as can be seen in Figure C-1. This poses some problems because all these elements are now coupled in their bending behaviour. This problem is discussed in Section 5-2-7.
- The same problem occurs at the connection points of the struts, the trailing edge wire and the canopy. They are all joined in one node, while the real situation is a bit more complex as can be seen in Figure C-1.
- The corner nodes of the kite will be used to connect the bridles to, called the bridle attachment points.

The main requirements for the mesh are stated as follows.

- The mesh should be as course as possible to obtain a fast model, but fine enough to reproduce the important deformation modes of the kite.
- The aerodynamic model that will be used is based on a mesh that is divided into five sections in the chordwise direction, so the finite element mesh will use this same division.
- The tips of the kite will be subjected to the largest deformations, are the most important for the steering behaviour and will be subjected to buckling effects. A finer mesh will be used for the tips to be able to catch all this behaviour and create a stable model.
- The division in span-wise direction should result in triangular elements with an aspect ratio (height/width of the triangular elements) that is close to one to maximize the accuracy of the elements.

A mesh generator was programmed in MATLAB with some parameters that can be varied to create the mesh as shown in Figure 5-10. This results in a model with the specifications as given in Table 5-1.

Table 5-1: Finite element properties of the kite

Parameter	Value
Beam elements	107
Shell elements	360
Nodes	222
DOF	1332

5-2-4 Canopy

Characteristics and material properties

The canopy generates the lift and drag forces and transfers them to the beams of the kite and acts as shear web between the struts and the leading edge and will be modelled with triangular shell elements.

The canopy is made of very thin ripstop material and undergoes large deformations when subjected to the aerodynamic loads. Measurements performed by EMPA materials science and technology [33] show that the thickness of the canopy is only $t = 0.08 \times 10^{-3}m$. They also show that ripstop is an anisotropic material and therefore the stiffness depends on the direction of the fibres. Mechanical properties based on plane stress linear elastic orhotropic model show a stiffness of $k_{warp} = 104.4kN/m$ in the warp direction and $k_{fill} = 71.2kN/m$ in the fill direction. Because modelling orthotropic material is difficult, the assumption is made that an average of $100kN/m$ can be used as to estimate the canopy stiffness. This shouldn't have a large effect, because the kite will also fly with different materials. This results in an Young's modulus of

$$E = \frac{100kN/m}{0.08 \times 10^{-3}m} = 1250MPa \tag{5-11}$$

Furthermore a Possion ratio of $\nu = 0.3$ will be used.

The ripstop material is unable to withstand compression forces and starts to wrinkle immediately when subjected to compression forces. This behaviour is not of interest in this research and makes computations slower, but is sometimes unavoidable in simulations as was discussed in Section 5-1-4. The canopy cannot withstand any real bending forces, but is merely just a membrane. Pure membrane elements are not commonly used in finite element analysis, because of their poor performance. But the bending stiffness in the element increases (decreases) with the thickness squared, while the in-plane stiffness varies linear with the thickness. The canopy is so thin that the bending stiffness almost disappears compared to the in-plane stiffness and the finite shell element should act in a realistic manner.

Non-linear three node triangular flat shell element

To model the canopy a three node shell element will be used that can handle the geometric non-linear behaviour. Since the development of a geometric non-linear finite surface element is difficult and a finite element programmed by Tiso [47] is available in MATLAB , its suitability for this thesis is investigated.

This three-node element has a membrane part for the in-plane forces and a bending part for the out-of-plane forces.

The linear bending contribution comes from the description of Allman [3]. He describes a triangular finite element for plate bending problems derived from the modified potential energy principle, using a cubic displacement field. The element uses three degrees of freedom per node, one translational and two rotational, which results in a total of 9 degrees of freedom for the element as can be seen in Figure 5-12. The degrees of freedom of the element can be described by

$$\boldsymbol{q}_{e,bending} = [w_1 \quad \theta_{x1} \quad \theta_{y1} \quad w_2 \quad \theta_{x2} \quad \theta_{y2} \quad w_3 \quad \theta_{x3} \quad \theta_{y3}]^T \tag{5-12}$$

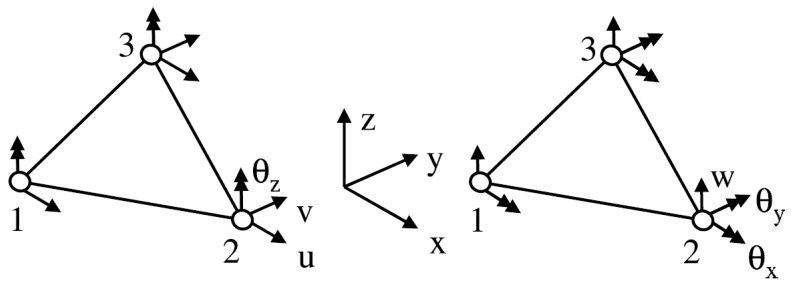


Figure 5-12: The triangular three node flat shell element with the membrane part (left) and the bending part (right) as described by Tiso [47].

The linear membrane (in-plane) contribution comes from the description in Allman [4] and Felippa [16] of the LST-3/9R element that is a member of the optimal Linear Strain Triangle (LST) family of elements, The Allmann '88 triangle. The three corner nodes have three

degrees of freedom, two in-plane translations and one drilling rotation, as depicted in Figure 5-12. The hierarchical drilling degrees of freedom are rotational degrees of freedom that are perpendicular to the plane of the triangle and a way to improve the performance of the element and express the higher order behaviour of the element. Cubic polynomial displacement functions are used to describe the displacement field u and v. This element calculates the basic and higher-order stiffness matrix and combines these as can be seen from Figure 5-13. The degrees of freedom of the element can be described by:

$$\boldsymbol{q}_{e,membrane} = [u_1 \quad v_1 \quad \theta_{z1} \quad u_2 \quad v_2 \quad \theta_{z2} \quad u_3 \quad v_3 \quad \theta_{z3}]^T \tag{5-13}$$

The membrane and bending element can be combined in one element that is capable of bending and in-plane deformations. This results in an element with 18 degrees of freedom:

$$\boldsymbol{q}_e = [u_1 \quad v_1 \quad w_1 \quad \theta_{x1} \quad \theta_{y1} \quad \theta_{z1} \quad u_2 \quad v_2 \quad w_2 \quad \theta_{x2} \quad \theta_{y2} \quad \theta_{z2} \quad u_3 \quad v_3 \quad w_3 \quad \theta_{x3} \quad \theta_{y3} \quad \theta_{z3}]^T \tag{5-14}$$

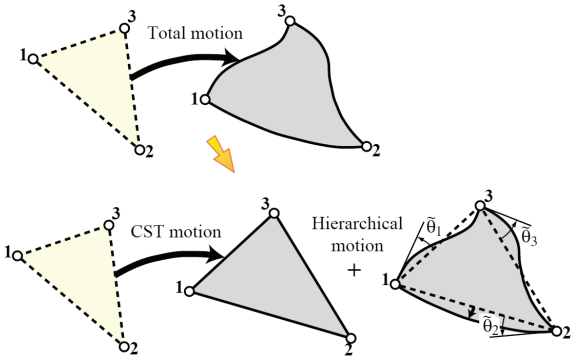


Figure 5-13: Hierarchical rotations due to the drilling degree of freedom Felippa [16].

The non-linear contributions come from the kinematic equations describing the axial stress ε and curvatures χ derived from the simplified Lagrangian strain tensor and neglecting shear effects, as described in detail by Tiso [47].

$$\begin{cases} \varepsilon_x & = \frac{1}{A}\int_A [u_{,x} + \frac{1}{2}(v_{,x}{}^2 + w_{,x}{}^2)]dA & \chi_{xx} & = \frac{1}{A}\int_A w_{,xx} dA \\ \varepsilon_y & = \frac{1}{A}\int_A [v_{,y} + \frac{1}{2}(u_{,y}{}^2 + w_{,y}{}^2)]dA & \chi_{yy} & = \frac{1}{A}\int_A w_{,yy} dA \\ \varepsilon_{xy} & = \frac{1}{A}\int_A [\frac{1}{2}(u_{,y} + y_{,x}) + \frac{1}{2}(w_{,x} w_{,y})]dA & \chi_{xy} & = \frac{1}{A}\int_A w_{,xy} dA \end{cases} \tag{5-15}$$

The element is based on isotropic linear material and the constitutive relations are described by

$$\boldsymbol{\sigma} = \boldsymbol{A}_m \boldsymbol{\epsilon} \tag{5-16}$$

with

$$\boldsymbol{A}_m = \frac{Eh}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{5-17}$$

The element formulation will return the elemental tangential stiffness matrix $\boldsymbol{K}_e(\boldsymbol{q}_e)$ and the internal force vector $\boldsymbol{f}_e(\boldsymbol{q}_e)$, both depending on the element displacements $\boldsymbol{q}_e$.

Validation

The linear and non-linear version of this element have been implemented in MATLAB and compared to the SHELL63 element in Ansys in Appendix B-2. In general the results in both cases are very similar. Only when a point load was applied, some differences were found. This was to be expected, because a point load is a difficult load case for a plate (acting on an infinite small area) and apparently the different elements handle this a bit different. Figure 5-14 shows that the model is capable of handling thin canopy with an extreme low stiffness and shows the expected deformations under an aerodynamic load. In general it can be said that this element is suitable for its purpose and therefore will be used in the modelling of the canopy of the kite.

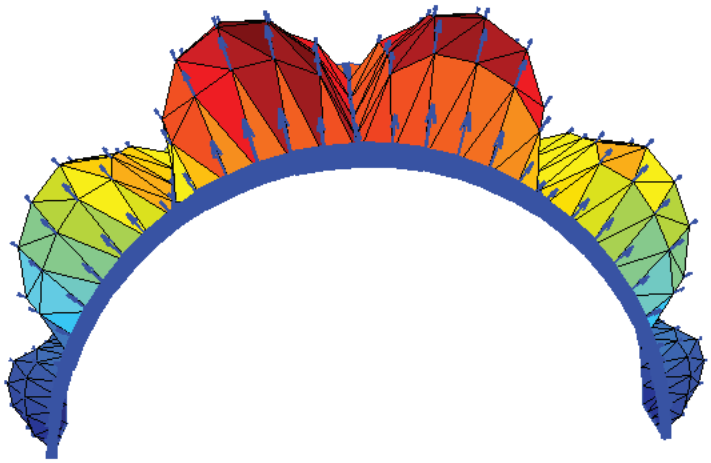


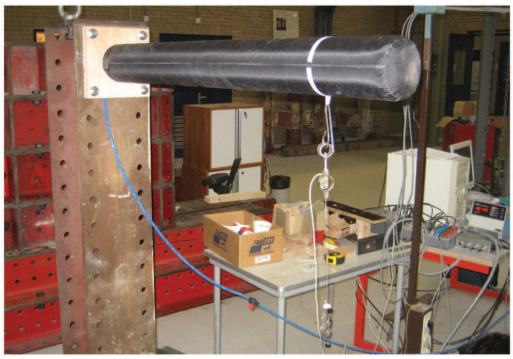
Figure 5-14: The kite with extreme compliant canopy material (less stiff than the real value) subjected to an aerodynamic load.

5-2-5 Leading edge and struts

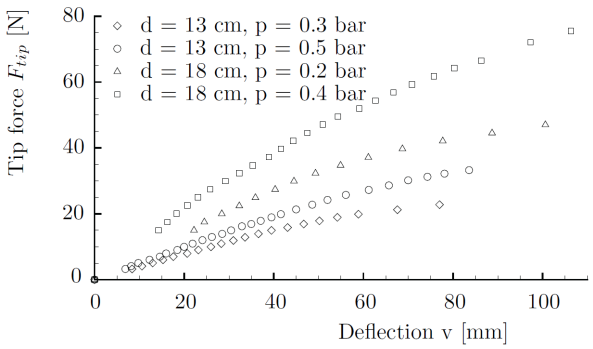
Characteristics and material properties

The inflatable leading edge and the struts form the backbone of the kite, they provide stiffness and transfer the lift and drag forces to the tethers. Their stiffness depends on the pressure inside, the diameter and deflection. The outer tube is made of the material Dacron and the inner tube is an airtight bladder from thermoplastic polyurethane (TPU). Veldman et al. [49] discusses the modelling problems of inflatable beams that come from wrinkling behaviour, collapses, modelling of the internal pressure field and non-linear bending. Thousands of surface elements are needed to model inflatable beams with finite surface elements to create a model that represents this behaviour in a realistic way. Since we are only interested in the global behaviour the option the model the inflatable beams with regular beams with similar material properties will be considered, thereby neglecting the wrinkling effects.

Breukels [7] performed bending and torsion experiments on inflatable beams to determine their stiffness properties. He used 1 meter beams with different diameters and a constant pressure and clamped them at one end as shown in Figure 5-15a. Loads were introduced at the end of the beam while measuring the deflection at the tip. The results of this experiment can be seen in Figure 5-15b.



(a) Set-up to measure the inflatable beam stiffness properties.



(b) Results of the inflatable beam stiffness measurements.

Figure 5-15: Experiments performed by Breukels [7] to measure the stiffness properties of inflatable beams.

The following function was fitted through the results with an R^2 fit of over 0.99 percent.

$$F_{tip} = f(p, r, v) = \left[(C_1 r + C_2)p^2 + (C_3 r^3 + C_4)\right]\left[1 - e^{-\frac{(C_5 r^5 + C_6)p + (C_7 r + C_8)}{(C_1 r + C_2)p^2 + (C_3 r^3 + C_4)} v}\right] \tag{5-18}$$

The domain where this function is valid is not completely clear, because it gives negative stiffnesses for beams with small radii and is therefore only considered to be valid between the radii 40mm-90mm, because all test beams were within that range. Some small radii in the kite will therefore be increased to 40mm. The function stops being valid when the aerodynamic forces become too high and inflatable beams collapse (see Figure 5-6). A steep drop in beam stiffness will be seen in that case. This is outside the scope of this thesis and will not be modelled, but this effect could be included in a later phase, because Breukels [7] also performed studies on this topic. The non linear material behaviour can clearly be seen from the Figure 5-15b, especially for lower internal pressures and higher deflections. These material characteristics are converted to values that can be used for regular beam elements.

Conventional beam theory yields:

$$v = \frac{F_{tip} L^3}{3EI} \tag{5-19}$$

$$EI = \frac{F_{tip} L^3}{3v} \tag{5-20}$$

where E is the Young's modulus, I the area moment of inertia, v the deflection and L the length of the beam. Combining these equations yields the bending stiffness for a beam element

$$EI(p, v, r) = \frac{F_{tip}(p, r, v) L^3}{3v} \tag{5-21}$$

Figure 5-16a shows this bending stiffness for a beam of 1m with a radius of 75mm to a varying deflection. It can be seen that the bending stiffness is not constant and varies with the deflection.

For a thin walled beam with a thickness t and round cross-sectional area the second moment of inertia is

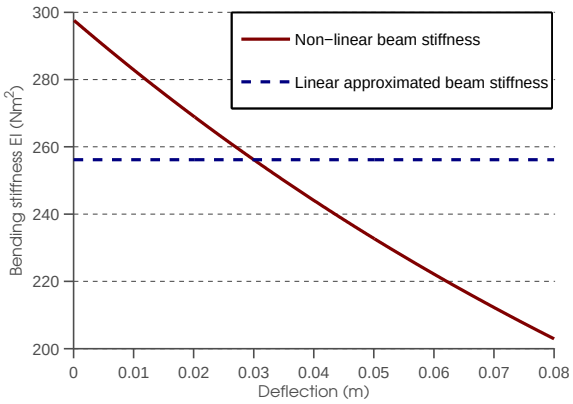
$$I = \frac{\pi(r^4 - (r-t)^4)}{4} \tag{5-22}$$

This leads after substituting in Equations 5-21 to the Young's modulus of a beam element

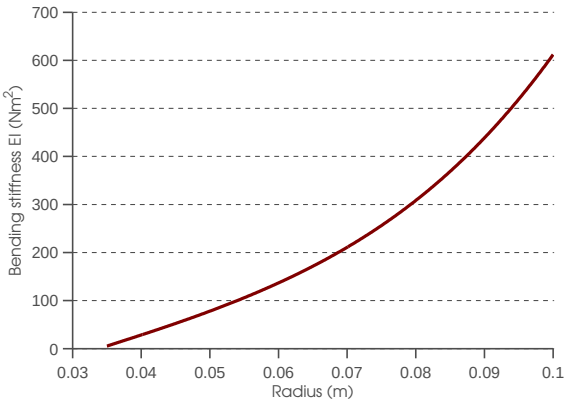
$$E(p, v, r) = \frac{4F_{tip}(p, r, v)L^3}{3vr^4} \tag{5-23}$$

This formula can be used to formulate the non-linear constitutive equations. The relation between strains and stresses for a beam element depend on the current deformation. A normal working pressure for the beams that will be used is a constant $p = 0.5bar$. In a first attempt to simulate the inflatable beams with regular beam element, the non-linear material behaviour will be approximated by assuming a fixed deflection of $v = 0.03m$. This will result in an underestimation of the stiffnes for low deflections and a overestimation of the stiffness for large deflections. This will slightly change the deflections of the whole kite, but tests with beams with different stiffnesses show that this has a minor effect compared to the influence of the aerodynamic forces. In a later phase the material non-linearity could be implemented.

Figure 5-16b shows how the bending stiffness of the beam varies with the radius of the beam when the deflection is fixed at $v = 0.03m$.



(a) Beam bending stiffness varying with the deflection for r=0.075m, t=2mm.



(b) Varying beam bending stiffness with the radius and a deflection for v=0.03m, t=2mm.

Figure 5-16: Stiffness of inflatable beams.

Geometric (non-)linear three dimensional two node beam element

A linear two node three dimensional Bernoulli beam without shear deflection was implemented in MATLAB and based on the description of Rixen [40].

Three dimensional geometric non linear beam elements are still a subject of research and it is not easy to find and element that can easily be applied and coded in MATLAB . Therefore a new three dimensional beam element with two nodes was developed that is able the do geometric non-linear analysis. The element as depicted in Figure 5-17 is based on a combined approach of Reddy [39] and Crisfield [10] and assumes small strains, small to moderate rotations and large displacements. It is a straight element, assuming that the round leading edge of the kite can be approximated by straight elements when the elements are small enough. The element is basically the non-linear version of the linear beam element that was implemented and is also based on the classic Bernoulli beam theory. It is assumed that the plane sections in the beam remain perpendicular to the axis of the beam and rigid after a deformation of the beam and therefore it is allowed to neglect the Poisson effect and transverse shear strains. Furthermore a linear torsion is assumed. The non-linearity comes from the inclusion of the in-plane forces that are proportional to the square of the rotation of the transverse normal to the beam axis and are included in the kinematic relation between the displacements and the strains. The axial strain ε and shear strain γ due to torsion are described by

$$\varepsilon_{11} = \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right) + \frac{1}{2}\left(\frac{\mathrm{d}v}{\mathrm{d}x}\right)^2 + \frac{1}{2}\left(\frac{\mathrm{d}w}{\mathrm{d}x}\right)^2 \tag{5-24}$$

$$\gamma = \rho\frac{\mathrm{d}\phi}{\mathrm{d}x} = \rho\kappa_x \tag{5-25}$$

with u, v and w the displacement fields and ρ the radial distance from the axis of the beam.

The virtual work principle is used to derive the internal forces and stiffness matrix. The shape functions for the torsion and axial displacement are taken linear and for the out of plane displacements cubic polynomial. The full derivation of the element is given in Appendix A. This element formulation returns the elemental tangential stiffness matrix $\boldsymbol{K}_e(\boldsymbol{q_e})$ and the internal force vector $\boldsymbol{f}_e(\boldsymbol{q_e})$, both depending on the element displacements $\boldsymbol{q}_e$. The element has a total of 12 degrees of freedom:

$$\boldsymbol{q}_{el}^T = \begin{bmatrix} u_1 & v_1 & w_1 & \psi_{x1} & \psi_{y1} & \psi_{z1} & u_2 & v_2 & w_2 & \psi_{x2} & \psi_{y2} & \psi_{z2} \end{bmatrix} \tag{5-26}$$

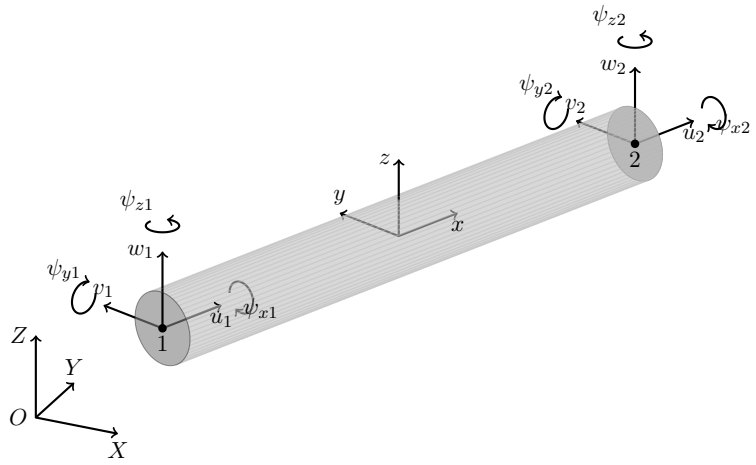


Figure 5-17: The geometric non-linear beam element.

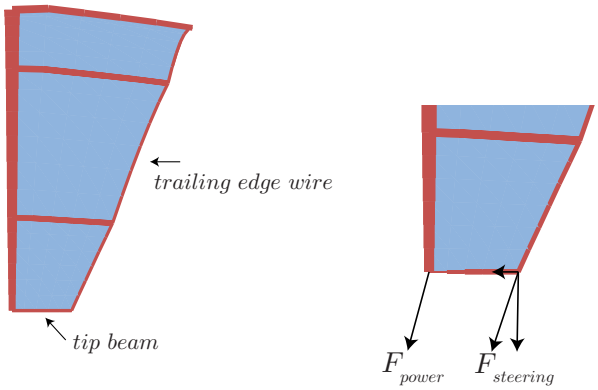
Validation

The linear and non-linear version of this element have been implemented in MATLAB and were compared to the BEAM4 element in Ansys in Appendix B-2. The MATLAB beam performed well and was slightly stiffer than the Ansys element, but besides that the results were similar. From these tests it was concluded that the element is suitable for this project.

5-2-6 Trailing edge wire

The kite has a thin trailing edge wire stitched in the trailing edge of the canopy to provide some extra stiffness, as showed in Figure 5-18a. This wire has two purposes. Firstly it prevents the trailing edge from fluttering, because the wire gives the trailing edge some bending stiffness when it is subjected to aerodynamic forces, forcing the trailing edge to stay in a nice round shape. Secondly it prevents the struts from bending to much upward and thereby retaining a nice aerodynamic profile and improving the aerodynamic performance.

This trailing edge wire will be added to the model by using the same beam elements that were used for the struts and the leading edge with a radius $r = 0.3mm$ and a Young's modulus of $E = 100GPa$ (Dyneema). This simulates a thin wire with low bending stiffness, but quite stiff in the axial direction. The trailing edge is connected to the last row of nodes that is used to model the canopy. At some nodes the trailing edge wire is connected to both the canopy and the end of the struts which couples the bending of the elements at those points. This bending coupling sometimes creates unwanted side-effects, but those are avoided by keeping the diameter of the wire small.



(a) Trailing edge wire and tip beam.

(b) Force in the tips due to a steering input.

Figure 5-18: Details of the tips, leading edge and trailing edge wire.

5-2-7 Tips

Figure C-2 and Figure 5-18a show that there is a thin, stiff beam in the tips to prevent the tips from extreme wrinkling/folding when a steering input is applied. The beam absorbs a part of the steering forces as depicted in Figure 5-18b.

This tip-beam will be modelled with the same beam elements as used to model the inflatable beams and connected to the last row of nodes that represent the tips. The bending of the leading edge, tip beams and trailing edge wire are now all coupled through the corner nodes. In reality this is not the case, because there is some canopy between them that will deform extremely. This coupled bending creates some problems. It can for instance result in an S-shaped bending of the tip. To circumvent this problem a tapered tip-beam is introduced that has a smaller radius near the connection between the leading edge and the tip-beam to create a more flexible connection between the tips and the leading edge that allows them to bend more independently of each other. Since the trailing edge wire is already very thin it can't transfer much bending, therefore the tip-beam will only be made thinner at the connection with the leading edge. The diameters of the tip-beam vary between 1mm at the connection point with the leading edge to 6mm at the trailing edge side. The beams are made from aluminium with a Young's modulus of $E = 69GPa$.

The tip-beam is basically a stiff spring in its axial direction. The dynamic simulation will 'see' this stiff spring, because the corners of the tip-beam are directly coupled to the bridles in the dynamic tether and bridles simulation. Stiff springs result in high frequency behaviour and can result in numerical instabilities and require very small time steps. There is no problem in the axial direction, because the axial movement will be constraint in the dynamic tether and bridles simulation. However, out-of-plane movements will not be constraint and the tip-beam also acts as a spring in those directions due to its bending behaviour. Therefore the tip-beam is modelled thinner than it is in reality, but still stiff enough to withstand the axial forces that come from the steering bridles. This reduces the bending stiffness of the beam and therefore removes the high-frequency behaviour.

5-2-8 External forces

The forces that act on the kite come from the aerodynamics, the structural damping, the bridles, the gravity and lateral and rotational accelerations. These forces are described in this section and all included in the external force vector $\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{X})$ as discussed in Section 5-1-2.

Aerodynamic forces

The most important and dominant applied external forces come from the aerodynamics. The used model of Breukels [7] will be discussed in detail in Chapter 6 and returns a distributed load $\boldsymbol{F}_{aero}$ for the kite based on the displacements of the kite $\boldsymbol{q}$, the velocity of the nodes $\boldsymbol{V}_{struct}$ in the structure, the velocity of the kite $\boldsymbol{V}_k$, the rotational velocity of the kite $\boldsymbol{\omega}_k$ and the wind $\boldsymbol{V}_w$ velocity.

$$\boldsymbol{F}_{aero} = f(\boldsymbol{q}, \boldsymbol{V}_{struct}, \boldsymbol{V}_k, \boldsymbol{\omega}_k, \boldsymbol{V}_w) \tag{5-27}$$

An example of the distributed forces is given in Figure 5-19.

Gravity forces

The gravitational forces are very small compared to the aerodynamic forces and adding them would have almost no effect on the deformation of the kite. The kite mass is in the order of

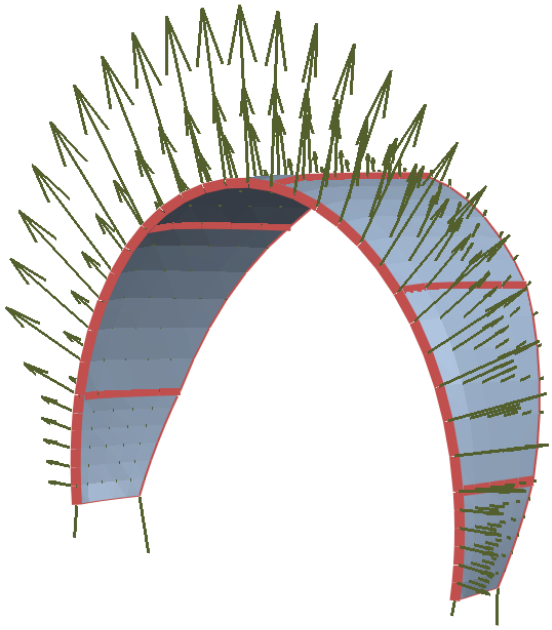


Figure 5-19: Distributed load from the aerodynamic model acting on the kite.

10kg and the generated lift forces in the order of 4000N when flying crosswind, resulting in a ratio of 1/40 between them. The mass of the canopy is so low, that the gravity won't have any effect on its deformation. The mass of the beams is a bit higher, but they are also stiffer, so the gravity has no effect on their deformation either.

The gravity does have influence on the global dynamics of the kite system. For instance, the obtained velocities when flying down are significantly higher than when flying up. Therefore the gravity of the kite will be incorporated in the dynamic tether and bridles model by including mass at the bridle attachment points (corner points of the kite) as will be discussed in Chapter 7.

Acceleration forces

Due to the quasi-static assumption of the kite as discussed in Chapter 4, the mass matrix $\boldsymbol{M}$ is neglected in the finite element equations and therefore also the accompanying acceleration forces. Other forces come from the lateral and rotational accelerations of the floating kite reference frame in the dynamic tether and bridles simulation. Results in Chapter 9 show that the maximum lateral speed of the kite is $30m/s$ and the maximum accelerations is 5 m/s^2. Resulting in a lateral acceleration force of 50N with a 10 kilo kite on a 100m line. The maximum force due to rotational acceleration becomes

$$F_{centrifugal} = \frac{mv^2}{r} = \frac{10 * 30^2}{100} = 90N \tag{5-28}$$

Both forces are small compared to the aerodynamic forces that are in the order of 4000N and will therefore be neglected.

Damping forces

The finite element method solves only static problems, due to the quasi-static assumption and therefore no damping forces are included. But damping turns out to be quite important to create a stable model. The proposed approach will result in a sequence of static solutions of the deformation of the kite, representing its dynamic behaviour. No applied damping will result in oscillations in the deformations and the prescribed displacements in the bridle attachment points sometimes result in sudden 'jumps' of the kite from one configuration to a completely different one with a total collapse as result.

In reality the transient movements of the kite damp out partly due to the internal structural damping, but mostly because of the aerodynamic drag forces as will be discussed in Chapter 6.

Rayleight like damping can be introduced to add some structural damping. A damping matrix $\boldsymbol{C}$ can be introduced that depends on the mass matrix $\boldsymbol{M}$ and the stiffness matrix $\boldsymbol{K}$ and the coefficients μ and λ. Since de mass matrix is not available, only the stiffness matrix term will be used.

$$\boldsymbol{C} = \mu \boldsymbol{M} + \lambda \boldsymbol{K} \tag{5-29}$$

The damping force is now determined by

$$\boldsymbol{F}_{damp,struct} = \boldsymbol{C}\boldsymbol{V}_{struct} \tag{5-30}$$

The structural damping forces will be seen as part of the external forces and not separately added to the equations of motions by writing the term $\boldsymbol{C}\dot{\boldsymbol{q}}$ which would normally be done. This is because $\dot{\boldsymbol{q}}$ and $\boldsymbol{V}_{struct}$ are not the same. $\dot{\boldsymbol{q}}$ would be calculated from a finite element time stepping scheme which is not used and $\boldsymbol{V}_{struct}$ is determined numerically from the current displacements $\boldsymbol{q}_n$ and the displacements at the previous time step $\boldsymbol{q}_{n-1}$ divided by the time step h.

$$\boldsymbol{V}_{struct} = \frac{\boldsymbol{q}_n + \boldsymbol{q}_{n-1}}{h} \tag{5-31}$$

$$\tag{5-32}$$

This numerical calculation from subsequent time steps always lacks one step behind, sometimes resulting in instabilities in the model if the coefficient λ is chosen too high. The aerodynamic damping also results in unstable models as will be discussed in Chapter 6. Therefore some additional numerical damping is added between the bridle attachment points in the dynamic tether and bridles model as will be discussed in Chapter 7.

5-2-9 Boundary conditions

The displacements of the four corner nodes of the kite, the bridle attachment points, are prescribed by the dynamic tether and bridles simulation and will be denoted by $\boldsymbol{q}_b$.

5-3 Solving the non-linear equations

An appropriate solution procedure is developed in this section to solve the non-linear algebraic finite element equations. The full solution procedure is depicted in Figure 5-24. Section 5-3-1

discusses the basic Newton-Raphson method to solve the structural non-linear equations. In Section 5-3-2 the solution procedure to solve the aeroelastic fluid-structure-interaction problem between the aerodynamic forces and the structural model is discussed. A load control algorithm is developed in Section 5-3-3 to improve the stability and convergence of the algorithm, followed by a discussion about the convergence criteria in Section 5-3-4. This section won't explain how the linear finite element equations are solved, this can be found in any introductory book on the finite elements method.

5-3-1 Structural solution procedure

The non-linear equations of the quasi-static kite that have to be solved were described by Equation 5-10 an repeated here for convenience.

$$\begin{cases} \boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{X}) \\ \boldsymbol{q}_0, \boldsymbol{q}_b, \boldsymbol{X} \qquad \textit{prescribed} \end{cases} \tag{5-33}$$

$\boldsymbol{f}(\boldsymbol{q})$ are the internal forces and $\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{X})$ are the external forces that depend on the displacements $\boldsymbol{q}$ and the prescribed state of the system $\boldsymbol{X}$ consisting of the kite velocity $\boldsymbol{V}_k$, the rotational kite velocity $\boldsymbol{\omega}_k$, the velocity of the structural nodes $\boldsymbol{V}_{struct}$ and the wind velocity $\boldsymbol{V}_w$. The initial displacements $\boldsymbol{q}_0$ are prescribed, either by the solution from the previous time step in the coupled simulation with the dynamic tether and bridles model or set to $\boldsymbol{q}_0 = \boldsymbol{0}$ when a single case is be solved. The boundary displacements $\boldsymbol{q}_b$ are prescribed by the dynamic tether and bridles model or set by hand to solve a single case.

The Newton-Raphson iteration scheme [40, 39, 14] can be used to solve these non-linear equations. This method is well suited for problems without difficulties in the equilibrium path (load-deflection curve) like multiple solutions, snap-through or buckling. During normal flight conditions these phenomena are not likely to occur very often, but they are also unavoidable as was described in Section 5-1-4.

The Newton-Raphson method is a member of the Newton-like solving algorithms with several variants for specific situations. The basic algorithm makes a prediction of the solution in the prediction phase, followed by an iterative correction phase to iterate to a balance between the internal and external forces to obtain the static solution to Equation 5-33.

First a prediction is made by saying that the new displacements $\boldsymbol{q}_{n+1}$ are the same as the displacements at the end of the last time step in the dynamic tether and bridles model $\boldsymbol{q}_n$

$$\boldsymbol{q}_n = \boldsymbol{q}_{n-1} \tag{5-34}$$

Solving the static equations has nothing to do with the time t_n, the time integration is handled by the dynamic tether and bridles model. However solving the finite element equations *will* result in the configuration at the next time step $n+1$. For clarity the subscript n will be omitted in the rest of this section and the newly calculated displacements are denoted by $\boldsymbol{q}$.

In the corrective phase, the residual force $\boldsymbol{r}$ of a configuration $\boldsymbol{q}$ is formulated by

$$\boldsymbol{r}(\boldsymbol{q}) = \boldsymbol{f}(\boldsymbol{q}) - \boldsymbol{g}(\boldsymbol{q}) \tag{5-35}$$

The algorithm tries to find a balance between the internal and external forces to make the residual zero $\boldsymbol{r} = \boldsymbol{0}$. The prescribed boundary conditions $\boldsymbol{q}_b$ will automatically be taken into

account here, since they cause internal forces as well.
Let us denote $\boldsymbol{q}^k$ as an approximate value for $\boldsymbol{q}$ resulting from structural iteration k. In the neighbourhood of this configuration, the residual can be estimated by the linear equation

$$\boldsymbol{r}_L(\boldsymbol{q}^{k+1}) = \boldsymbol{r}_L(\boldsymbol{q}^k + \Delta\boldsymbol{q}^k) \simeq \boldsymbol{r}(\boldsymbol{q}^k) + \boldsymbol{S}(\boldsymbol{q}^k)\Delta\boldsymbol{q}^k = \boldsymbol{0} \tag{5-36}$$

with the iteration matrix

$$\boldsymbol{S} = \left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}}\right]_{\boldsymbol{q}^k} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} - \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}} \tag{5-37}$$

The matrix $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}$ describes how the external forces vary with the displacements. However, the gradient information of the aerodynamic forces is not available and the resulting matrix would not be symmetric. Because the solution will also converge without this matrix and to preserve the symmetry of the iteration matrix, this matrix is omitted.

The matrix $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}$ represents the tangential stiffness matrix $\boldsymbol{K}_t$ which is to be recalculated for every iteration.

The equation can now be solved iteratively for the free degrees of freedom in $\boldsymbol{q}$ by correcting the solution $\boldsymbol{q}$ at every iteration step with $\Delta\boldsymbol{q}^k$ as shown in Figure 5-20.

$$\boldsymbol{K}_t(\boldsymbol{q}^k)\Delta\boldsymbol{q}^k = -\boldsymbol{r}(\boldsymbol{q}^k) \tag{5-38}$$

$$\boldsymbol{q}^{k+1} = \boldsymbol{q}^k + \Delta\boldsymbol{q}^k \tag{5-39}$$

The iteration process stops when the following criteria is satisfied which says that the residual should be small enough compared to the internal forces.

$$\frac{\left\|\boldsymbol{r}(\boldsymbol{q}^k)\right\|}{\|\boldsymbol{f}(\boldsymbol{q}^k)\|} < \varepsilon_{fem} \tag{5-40}$$

The final step is to obtain the boundary forces at the bridle attachment points, because those are exerted on the endpoints of the bridles in the dynamic tether and bridles model. The external forces at the bridle attachment nodes $\boldsymbol{F}_b$ are equal to the internal forces at the boundary nodes $\boldsymbol{f}_b$ and simply become

$$\boldsymbol{F}_b = \boldsymbol{f}_b \tag{5-41}$$

This altogether leads to Algorithm 1 and corresponds to the blue part in Figure 5-24.

5-3-2 Aero-elastic solution procedure

The external forces $\boldsymbol{g}$ and structural displacements $\boldsymbol{q}$ both depend on each other and form a coupled aero-elastic fluid-structure-interaction problem together.

In Algorithm 1 the solution procedure starts from a certain initial configuration $\boldsymbol{q}^0$ and calculates the external forces for those displacements $\boldsymbol{g}(\boldsymbol{q}^0, \boldsymbol{X})$. The external forces are updated in every iteration step with the new displacements $\boldsymbol{g}(\boldsymbol{q}^{k+1}, \boldsymbol{X})$.

This algorithm will result in an unstable system because the aerodynamic forces are very sensitive to changes in the displacements of the kite. In the first iteration steps, going from an

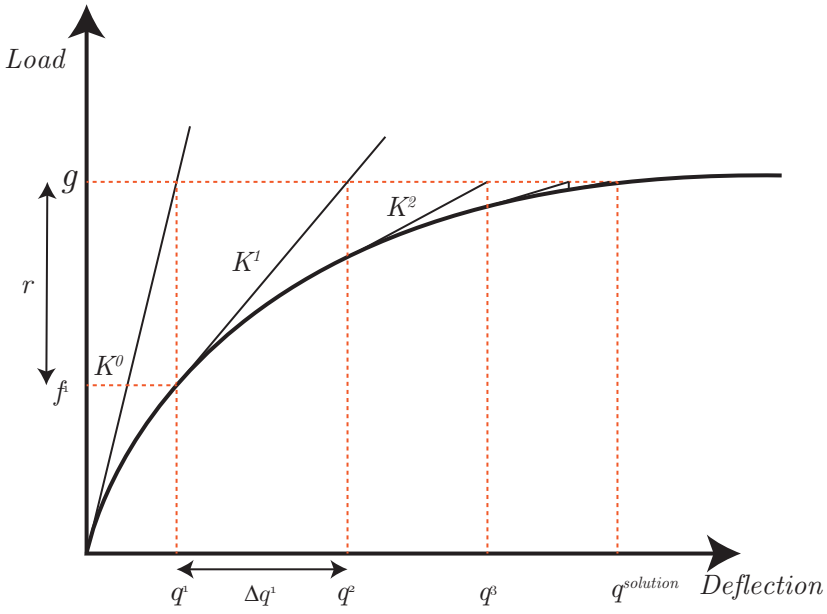


Figure 5-20: The Newton-Raphson method.

undeformed to a deformed kite, the stiffness of the canopy in the direction of the aerodynamic forces is so low that the displacements become very large resulting in very high (unrealistic) aerodynamic forces. This results in a very slow convergence or even un unstable system.

Therefore the iteration algorithm is split into two loops: a structural convergence loop (blue) and a aerodynamic convergence loop (green) as depicted in Figure 5-24. The external forces are kept fixed during the structural Newton-Raphson iterations. After obtaining the new structural converged solution the new external forces are calculated and a new structural convergence iteration starts. $\boldsymbol{q}^k$ denotes an approximate value of the displacements $\boldsymbol{q}$ after structural convergence step k and $\tilde{\boldsymbol{q}}^{\boldsymbol{m}}$ denotes the approximation of $\boldsymbol{q}$ after the aerodynamic convergence step m. $\boldsymbol{g}^m$ are the external forces in aerodynamic iteration step m.

The aerodynamic iterations stop when the external forces and displacements don't change any more. Two functions are defined that calculate the difference in configuration and external forces after a new aerodynamic iteration as

$$\boldsymbol{r}_{a,1} = \tilde{\boldsymbol{q}}^{m+1} - \tilde{\boldsymbol{q}}^{m} \tag{5-42}$$

$$\boldsymbol{r}_{a,2} = \boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1}, \boldsymbol{X}) - \boldsymbol{g}(\tilde{\boldsymbol{q}}^{m}, \boldsymbol{X}) \tag{5-43}$$

$\boldsymbol{r}_{a,1}$ and $\boldsymbol{r}_{a,2}$ should be small enough compared to respectively the displacement vector and external forces to consider the solution to be converged.

$$c_{a,1} = \frac{\|\boldsymbol{r}_{a,1}\|}{\|\tilde{\boldsymbol{q}}^{m+1}\|} < \varepsilon_{aero,1} \tag{5-44}$$

$$c_{a,2} = \frac{\|\boldsymbol{r}_{a,2}\|}{\|\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1})\|} < \varepsilon_{aero,2} \tag{5-45}$$

This creates a very rudimentary optimizing algorithm that can end up in a situation where it 'jumps' between two configurations, as illustrated in Figure 5-21. The initial displacements

Algorithm 1 Basic Newton-Raphson

Initialize structural iteration $k = 0$, boundary conditions $\boldsymbol{q}_b$ and state vector $\boldsymbol{X}$ from the dynamic simulation

Predictor $\boldsymbol{q}^k = \boldsymbol{q}_{n-1}$

Calculate internal and external forces
$\boldsymbol{g}(\boldsymbol{q}^k)$, $\boldsymbol{f}(\boldsymbol{q}^k, \boldsymbol{X})$

Calculate forces residual
$\boldsymbol{r}(\boldsymbol{q}^k) = \boldsymbol{f}(\boldsymbol{q}^k) - \boldsymbol{g}(\boldsymbol{q}^k, \boldsymbol{X})$

while $\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{f}\|} > \varepsilon$ **do**
  Corrector
  $\Delta\boldsymbol{q}^k = -\boldsymbol{K}_t^{-1}(\boldsymbol{q}^k)\boldsymbol{r}(\boldsymbol{q}^k)$
  $\boldsymbol{q}^{k+1} = \boldsymbol{q}^k + \Delta\boldsymbol{q}^k$

  Recalculate internal and external forces
  $\boldsymbol{g}(\boldsymbol{q}^{k+1}, \boldsymbol{X})$, $\boldsymbol{f}(\boldsymbol{q}^{k+1})$

  Recalculate forces residual
  $\boldsymbol{r}(\boldsymbol{q}^{k+1}) = \boldsymbol{f}(\boldsymbol{q}^{k+1}) - \boldsymbol{g}(\boldsymbol{q}^{k+1}, \boldsymbol{X})$

  $k = k + 1$
end while
$\boldsymbol{F}_b = \boldsymbol{f}_b$

$\tilde{\boldsymbol{q}}^0$ produce the aerodynamic force $\boldsymbol{g}^0$. The new calculated displacements after the structural iteration process are $\tilde{\boldsymbol{q}}^1$ and result in the new forces $\boldsymbol{g}^1$. These forces bring the displacements after structural convergence to $\tilde{\boldsymbol{q}}^2$, which are almost the same displacements as $\tilde{\boldsymbol{q}}^0$. This will never, or very slowly, converge to the solution point $(\tilde{\boldsymbol{q}}^{solution}, \boldsymbol{g}^{solution})$.

A simple algorithm is introduced to solve this problem and speed-up the convergence. The algorithm incrementally increases the aerodynamic forces into the new direction. It updates the old external forces $\boldsymbol{g}^m$ in the direction of the new calculated external forces $(\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1}) - \boldsymbol{g}(\tilde{\boldsymbol{q}}^m))$ with the parameter α to obtain the external forces in the next aerodynamic iteration step $\boldsymbol{g}^{m+1}$

$$\boldsymbol{g}^{m+1} = \boldsymbol{g}^m + \alpha(\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1}) - \boldsymbol{g}(\tilde{\boldsymbol{q}}^m)) \tag{5-46}$$

The parameter α is updated after every aerodynamic iteration. If the two residuals become smaller, the iteration was successful and α is increased, but if the residuals increase the iteration was not successful and α is decreased and the new obtained solution $\tilde{\boldsymbol{q}}^{m+1}$ will not be used and the iteration is redone with a smaller α parameter. This altogether leads to Algorithm 2 and is represented by the green block in Figure 5-24.

The advantage of this split between structural and aerodynamic convergence is that the solution method becomes more robust. There is less chance that the procedure won't converge

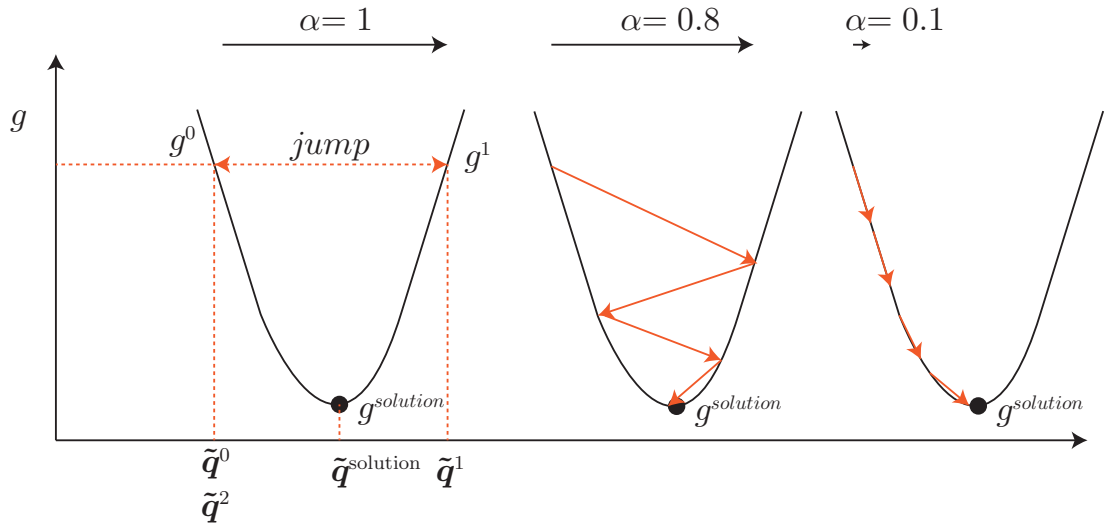


Figure 5-21: Aerodynamic convergence problem. The algorithm can get stuck between two static solutions (left). The α parameter was introduced to walk stepwise to the final solution.

since errors in the structural convergence won't lead to larger errors in the aerodynamic model that cause the solution procedure to become unstable.

5-3-3 Load control, stability and convergence procedures

A load control algorithm is implemented to speed up the convergence and improve the stability. The system won't converge without this load control algorithm.

The problem is depicted in Figure 5-22. The displacements in the first structural iteration step $\boldsymbol{q}^1$ become very large when an undeformed kite is suddenly exposed to the aerodynamic forces. The canopy has no stiffness in the out of plane direction, but the axial stiffness will withstand the aerodynamic forces when the canopy deforms. This principle was illustrated in Figure 5-4. The first iteration step largely overshoots the solution, resulting in a very unrealistic large displacement of the canopy. This process will lead to a very bad convergence or no convergence at all.

This phenomenon won't occur as much in normal flight conditions when the kite model is coupled to the dynamic tether and bridles models, because the solution process in every time step starts from an initially deformed kite and results in a slightly differently deformed kite. The initial tangential stiffness matrix has already stiffness developed in the direction of the aerodynamic forces. The problems occurs mainly in the start-up phase of the full simulation, or when studying single cases or when sudden large changes in flight conditions occur.

The performance can be improved by applying load control as described by Felippa [14]. The load is increased in a stepwise manner, allowing the structure to develop stiffness in the out-of-plane direction due to the bending of the canopy. This also prevents the solution process from getting stuck in a certain configuration. When the convergence fails, the subsequent results of the increments can be viewed by an engineer to gain more insight in the structural behaviour to see where the problem is. This method improves the robustness of the process and increases the speed. Even though smaller load steps have to be taken, less iterations per

Algorithm 2 Newton-Raphson with aero-elasticity

Initialize aerodynamic iteration $m = 0$, boundary conditions $\boldsymbol{q}_b$ and state vector $\boldsymbol{X}$, initial configuration $\tilde{\boldsymbol{q}}^m = \boldsymbol{q}_{n-1}$

Calculate external forces and aerodynamic convergence criteria
$\boldsymbol{g}^m = \boldsymbol{g}(\tilde{\boldsymbol{q}}^m, \boldsymbol{X})$
$c_{a,1}^m = c_{a,2}^m = inf$
$\alpha = \alpha_{init}$

while $c_{a,1}^m > \varepsilon_1$ AND $c_{a,2}^m > \varepsilon_2$ **do**
Initialize structural iteration $k = 0$
Predictor $\boldsymbol{q}^k = \tilde{\boldsymbol{q}}^m$

Calculate the internal forces and residual
$\boldsymbol{f}(\boldsymbol{q}^k)$
$\boldsymbol{r}(\boldsymbol{q}^k) = \boldsymbol{f}(\boldsymbol{q}^k) - \boldsymbol{g}^m$

while $\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{f}\|} > \varepsilon_3$ **do**
Corrector
$\Delta \boldsymbol{q}^k = -\boldsymbol{K}_t^{-1}(\boldsymbol{q}^k)\boldsymbol{r}(\boldsymbol{q}^k)$
$\boldsymbol{q}^{k+1} = \boldsymbol{q}^k + \Delta \boldsymbol{q}^k$

Recalculate internal forces and residual
$\boldsymbol{f}(\boldsymbol{q}^{k+1})$
$\boldsymbol{r}(\boldsymbol{q}^{k+1}) = \boldsymbol{f}(\boldsymbol{q}^{k+1}) - \boldsymbol{g}^m$
$k = k + 1$
end while
$\tilde{\boldsymbol{q}}^{m+1} = \boldsymbol{q}^k$

Recalculate external forces and aerodynamic convergence criteria
$\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1}, \boldsymbol{X})$
$c_{a,1}^m = \frac{\|\tilde{\boldsymbol{q}}^{m+1} - \tilde{\boldsymbol{q}}^m\|}{\|\tilde{\boldsymbol{q}}^{m+1}\|}$
$c_{a,2}^m = \frac{\|\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1}, \boldsymbol{X}) - \boldsymbol{g}(\tilde{\boldsymbol{q}}^m, \boldsymbol{X})\|}{\|\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1})\|}$
if $c_{a,1}^{m+1} >= c_{a,1}^m$ OR $c_{a,2}^{m+1} >= c_{a,2}^m$ **then**
$\alpha =$decrease α
$\boldsymbol{g}^{m+1} = \boldsymbol{g}^{m-1} + \alpha(\boldsymbol{g}(\tilde{\boldsymbol{q}}^m) - \boldsymbol{g}(\tilde{\boldsymbol{q}}^{m-1}))$
$\tilde{\boldsymbol{q}}^{m+1} = \tilde{\boldsymbol{q}}^m$
else
$\alpha =$increase α
$\boldsymbol{g}^{m+1} = \boldsymbol{g}^m + \alpha(\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1}) - \boldsymbol{g}(\tilde{\boldsymbol{q}}^m))$
end if
$m = m + 1$
end while
$\boldsymbol{F}_b = \boldsymbol{f}_b$

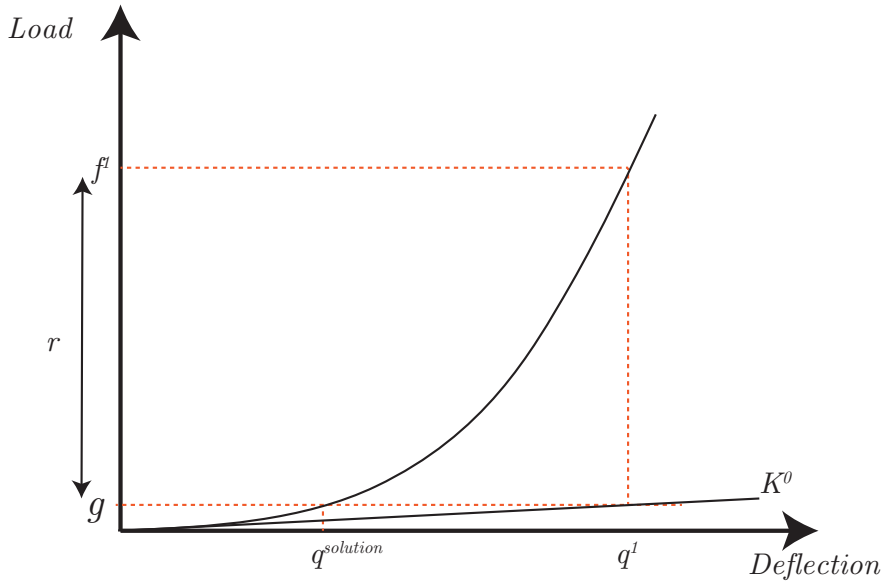


Figure 5-22: Slow convergence of the Newton-Raphson method for the kite. The undeformed canopy of the kite has a very low stiffness in the direction of the aerodynamic forces, resulting in an slow convergence process.

load step are needed and the configuration won't jump between two configurations. An smart algorithm that continuously adjusts the load step-size also speeds up the convergence.

A natural way to perform the load stepping is to slowly increase the inputs to the aerodynamic model from the old inputs to the new inputs. Chapter 6 describes that the inputs are the displacements $\boldsymbol{q}$, the wind velocity $\boldsymbol{V}_w$, the kite velocity $\boldsymbol{V}_k$, the rotational kite velocity $\boldsymbol{\omega}_k$ and the velocity of the nodes (structural speed) $\boldsymbol{V}_{struct}$ all combined in the vector $\boldsymbol{X}$. The prescribed boundary displacements $\boldsymbol{q}_b$ are also taken into account in the load control algorithm, because they also cause a part of the load.

The inputs are step-wise increased by multiplying them by a pseudo-time factor λ_s at pseudo-time step s that is 0 of no new load is applied and 1 if the full new load is applied. $\boldsymbol{X}_s$ and $\boldsymbol{q}_{b,s}$ denote the values at pseudo-time step s and $\boldsymbol{X}^0$ and $\boldsymbol{q}_b^0$ are the values at the previous time-step from the dynamic tether and bridles model and $\boldsymbol{X}^1$ and $\boldsymbol{q}_b^1$ are the new inputs from the dynamic model. The inputs at each pseudo-time-step can be written as

$$\boldsymbol{X}_s = \boldsymbol{X}^0 + \lambda_s(\boldsymbol{X}^1 - \boldsymbol{X}^0) \tag{5-47}$$

$$\boldsymbol{q}_{b,s} = \boldsymbol{q}_b^0 + \lambda_s(\boldsymbol{q}_b^1 - \boldsymbol{q}_b^0) \tag{5-48}$$

This means that the inputs to the aerodynamic model and prescribed displacements slowly increase from the last known situation (results from the previous time step in the dynamic tether model) to the new situation.

A guess for the initial pseudo-time step $\Delta\lambda_s$ has to be made. If the initial step size is chosen too low, time is lost by making too many small steps. If the initial step size is chosen too large, time is lost in restarting the iteration process with a smaller step size. A very small pseudo-time step size $\Delta\lambda_s$ has to be chosen when the initial configuration is undeformed. When coupled with the dynamic simulation, an initial guess of the step size is based on the step-size used in the precious dynamic time step.

A new step size can be chosen after each pseudo-time-step s, based on the needed number of structural k and aerodynamic m iterations. This has to be done efficiently to obtain the highest simulation speed. When the solution converged very fast, the step size can be increased and vice versa. When the solution doesn't converge within N_1 iterations the process is restarted with a smaller time step. The new step-size $\Delta\lambda_{s+1}$ is determined by

$$\mu_\lambda = (a + b * l)^c + d \tag{5-49}$$

$$\Delta\lambda_{s+1} = \mu_\lambda \Delta\lambda_s \tag{5-50}$$

where μ_λ is a multiplier, l the total number of structural iterations and aerodynamic iterations needed in one pseudo time step and a,b,c,d coefficients to be chosen.

The values of a,b,c,d are chosen based on the stability and stiffness of the used kite model. Stiffer models can use larger time-steps. The characteristics as displayed in Figure 5-23 are used in this thesis and were obtained from experiments with the model. In normal flight conditions, most time steps can be solved in just one pseudo-time step, because the deformations will only slightly change. Therefore the influence of the multiplier μ_λ on the overall speed of the simulation is low. The implementation of the algorithm improved the robustness of the kite model very much.

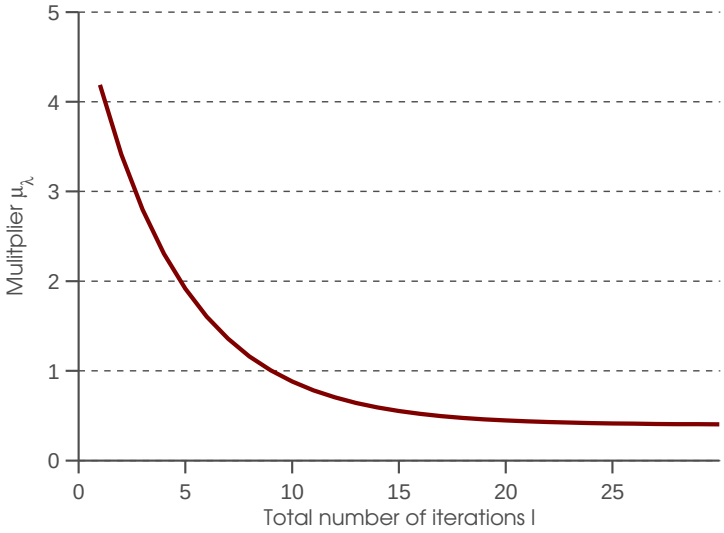


Figure 5-23: The characteristics of the pseudo time step multiplier μ_λ for the coefficients a=9.998, b=2.220, c=-1.031, d=3.982.

This altogether leads to Algorithm 3 and is graphically shown in Figure 5-24 in the orange part. $\bar{\boldsymbol{q}}^s$ denotes an approximation of $\boldsymbol{q}$ after load step s.

5-3-4 Convergence criteria

There are three convergence criteria in the convergence process. One for the structural convergence, as shown in Equation 5-40 and two for the aerodynamic convergence as shown in Equations 5-44. Stricter criteria lead in general to longer simulation times, because more

Algorithm 3 Newton-Raphson with aero-elasticity and load control

Initialize boundary conditions $\boldsymbol{q}_b^1$, $\boldsymbol{q}_b^0$ and state vector $\boldsymbol{X}^1$, $\boldsymbol{X}^0$, initial configuration $\bar{\boldsymbol{q}}^s = \boldsymbol{q}_{n-1}$, pseudo time-step $s = 1$, pseudo time $\lambda_s = \Delta\lambda_s$
while $\lambda_s <= 1$ AND $\lambda_{s-1} \neq 1$ **do**
  Initialize iteration counters $l = 0$, $m = 0$

  Calculate external forces and aerodynamic convergence criteria
  $\boldsymbol{X}_s = \boldsymbol{X}^0 + \lambda_s(\boldsymbol{X}^1 - \boldsymbol{X}^0)$
  $\boldsymbol{q}_{b,s} = \boldsymbol{q}_b^0 + \lambda_s(\boldsymbol{q}_b^1 - \boldsymbol{q}_b^0)$
  $\boldsymbol{g}^m = \boldsymbol{g}(\bar{\boldsymbol{q}}^s, \boldsymbol{X}_s)$
  $\tilde{\boldsymbol{q}}^m = \bar{\boldsymbol{q}}^s$
  $c_{a,1}^m = c_{a,2}^m = inf$
  $\alpha = \alpha_{init}$

  while $c_{a,1}^m > \varepsilon_1$ AND $c_{a,2}^m > \varepsilon_2$ **do**
    Initialize structural iteration $k = 0$
    Predictor $\boldsymbol{q}^k = \tilde{\boldsymbol{q}}^m$

    Calculate the internal forces and residual
    $\boldsymbol{f}(\boldsymbol{q}^k)$
    $\boldsymbol{r}(\boldsymbol{q}^k) = \boldsymbol{f}(\boldsymbol{q}^k) - \boldsymbol{g}^m$

    while $\frac{\|\boldsymbol{r}\|}{\|\boldsymbol{f}\|} > \varepsilon_3$ **do**
      Corrector
      $\Delta\boldsymbol{q}^k = -\boldsymbol{K}_t^{-1}(\boldsymbol{q}^k)\boldsymbol{r}(\boldsymbol{q}^k)$
      $\boldsymbol{q}^{k+1} = \boldsymbol{q}^k + \Delta\boldsymbol{q}^k$

      Recalculate internal forces and residual
      $\boldsymbol{f}(\boldsymbol{q}^{k+1})$
      $\boldsymbol{r}(\boldsymbol{q}^{k+1}) = \boldsymbol{f}(\boldsymbol{q}^{k+1}) - \boldsymbol{g}^m$
      $k = k + 1$, $l = l + 1$
    end while
    $\tilde{\boldsymbol{q}}^{m+1} = \boldsymbol{q}^k$

    Recalculate external forces and aerodynamic convergence criteria
    $\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1}, \boldsymbol{X}_s)$
    $c_{a,1}^m = \frac{\|\tilde{\boldsymbol{q}}^{m+1} - \tilde{\boldsymbol{q}}^m\|}{\|\tilde{\boldsymbol{q}}^{m+1}\|}$
    $c_{a,2}^m = \frac{\|\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1}, \boldsymbol{X}_s) - \boldsymbol{g}(\tilde{\boldsymbol{q}}^m, \boldsymbol{X}_s)\|}{\|\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1})\|}$
    if $c_{a,1}^{m+1} >= c_{a,1}^m$ OR $c_{a,2}^{m+1} >= c_{a,2}^m$ **then**
      α =decrease α
      $\boldsymbol{g}^{m+1} = \boldsymbol{g}^{m-1} + \alpha(\boldsymbol{g}(\tilde{\boldsymbol{q}}^m) - \boldsymbol{g}(\tilde{\boldsymbol{q}}^{m-1}))$
      $\tilde{\boldsymbol{q}}^{m+1} = \tilde{\boldsymbol{q}}^m$
    else
      α =increase α
      $\boldsymbol{g}^{m+1} = \boldsymbol{g}^m + \alpha(\boldsymbol{g}(\tilde{\boldsymbol{q}}^{m+1}) - \boldsymbol{g}(\tilde{\boldsymbol{q}}^m))$
    end if
    $m = m + 1$, $l = l + 1$
  end while
  $\bar{\boldsymbol{q}}^{s+1} = \tilde{\boldsymbol{q}}^m$
  Update pseudo time λ_s according to Algorithm 4
  $s = s + 1$
end while
$\boldsymbol{F}_b = \boldsymbol{f}_b$

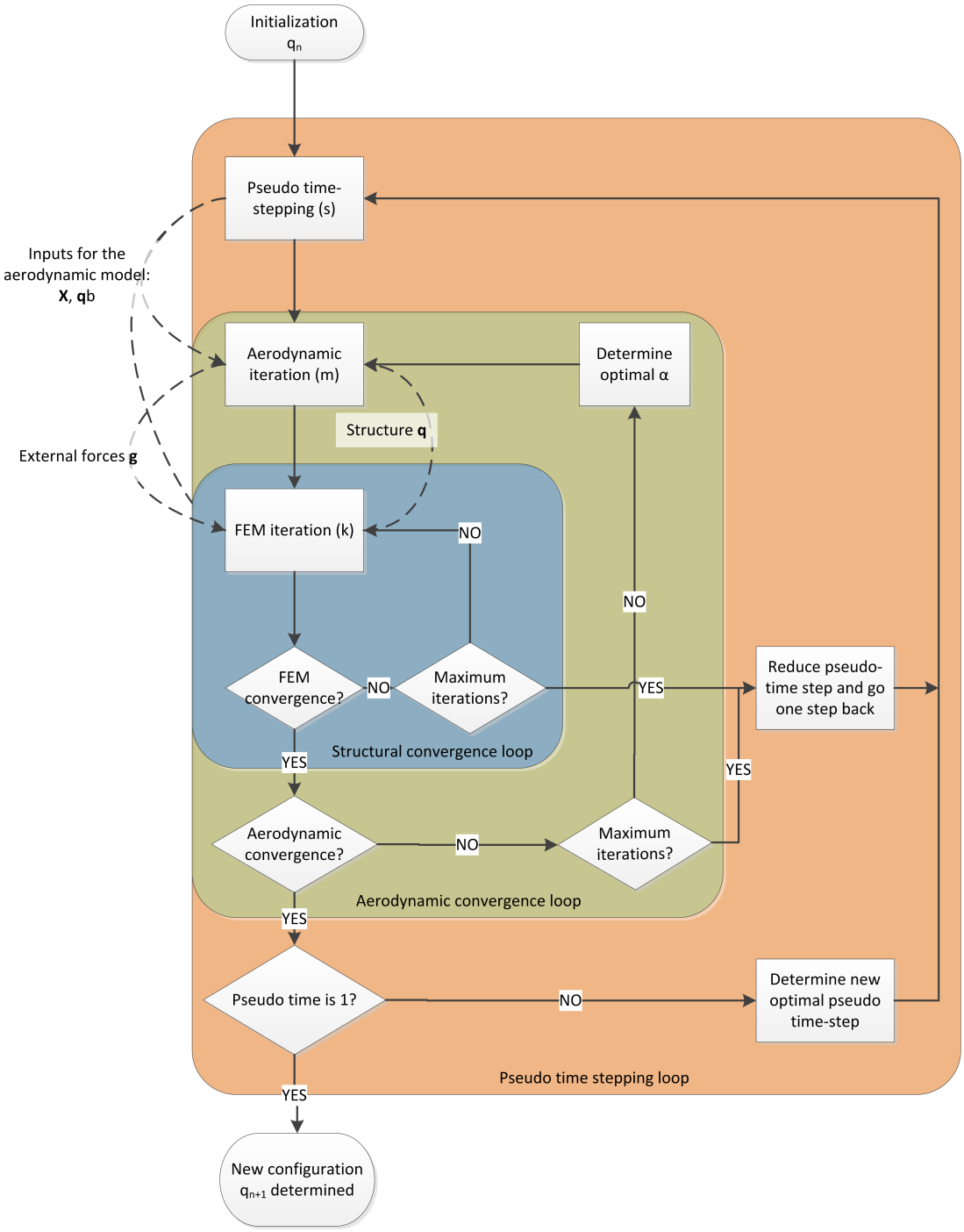


Figure 5-24: Graphical representation of the finite element solution procedure.

Algorithm 4 Pseudo time updating algorithm

Update pseudo time
if $l == N_1$ **then**
  Not converged, go one step back with a smaller pseudo time-step
  $s = s - 1$
  $\Delta\lambda_{s+1} = \Delta\lambda_s/2$
else
  Increase the pseudo time-step
  $\mu_\lambda = (a + b * l)^c + d$
  $\Delta\lambda_{s+1} = \mu_\lambda \Delta\lambda_s$
end if
if $\Delta\lambda_{s+1} > \Delta\lambda_{max}$ **then**
  $\Delta\lambda_{s+1} = \Delta\lambda_{max}$
end if
if $\lambda_s + \Delta\lambda_{s+1} > 1$ **then**
  $\Delta\lambda_{s+1} = 1 - \lambda_s$
  $\lambda_{s+1} = 1$
else
  $\lambda_{s+1} = \lambda_s + \Delta\lambda_{s+1}$
end if

iterations are needed. The criteria can be set rather low in this research, because we are not interested in high accuracies, but too small criteria can lead to instabilities.

Another advantage of using low criteria, is that it allows the simulation to proceed when it gets slow because it tries so solve some small buckling/wrinkling issues in the canopy, behaviour that is not of our interest. By setting the convergence criteria low enough, the convergence algorithm just steps over these problem without a complete convergence.

The speed of the convergence is further increased by using lower convergence criteria for the intermediate pseudo time steps and use strict criteria only for the last pseudo time step.

The values that were chosen to work well for this problem are:

$$\epsilon_{fem,end} = 1 \times 10^{-3} \tag{5-51}$$
$$\epsilon_{fem,intermediate} = 5 \times 10^{-2} \tag{5-52}$$
$$\epsilon_{aero1,end} = 1 \times 10^{-2} \tag{5-53}$$
$$\epsilon_{aero1,intermediate} = 5 \times 10^{-5} \tag{5-54}$$
$$\epsilon_{aero2,end} = 5 \times 10^{-2} \tag{5-55}$$
$$\epsilon_{aero2,intermediate} = 2 \times 10^{-1} \tag{5-56}$$

If the algorithm exceeds a certain number of maximum iterations or if the pseudo time-step becomes too small $\Delta\lambda_s < 1e{-}5$ the simulation stops.

5-4 Concluding remarks

This chapter described the finite element modelling of a flexible kite. The non-linearities of the kite were discussed and it was shown that it is necessary to included them in the derivation of the non-linear quasi-static finite element equations. The canopy was coarsely meshed with triangular shell elements and the inflatable beams with regular beam elements with representative material properties. Also the specific modelling of difficult areas as the tips and leading edge wire were discussed. This resulted in a reduced finite element model with a minimum amount of degrees of freedom that is able to show the global behaviour of the kite.

A solving algorithm was introduced to efficiently solve the aero-elastic fluid-structure-interaction problem by separating the solving process in a structural convergence and aerodynamic converge loop. A load control algorithm was developed to make the solution process more robust and speed up the convergence.

Problems arise when difficult behaviour as buckling/folding/wrinkling of the canopy occurs. This behaviour is not of our interest but unavoidable in some flight conditions. The solving algorithm was adjusted to deal with these situations as good as possible, although the model will still fail when the deformations or buckling gets too large.

Introducing the correct damping in the model poses some problems. The damping from the aerodynamic model results in instabilities as will be discussed in the next Chapter. Too much structural damping also results in instabilities, because the speeds of the degrees of freedom are not explicitly calculated in the finite element solution process due to the quasi-static assumption but need to be determined from the previous time-steps in the dynamic tether and bridles model.

The implementation of the elements has been verified and shows accurate results. Verification of the implementation of the whole kite will be further discussed in Chapter 9. It will be shown that the model performs well and shows the non-linear global deformations realistically.

Chapter 6

Aerodynamic model

The aerodynamic model that will be implemented in this thesis is the model developed by Breukels [7] as was discussed in Chapter 4. This chapter describes the basic idea of the model in Section 6-1, followed by the implementation in the finite element kite model in Section 6-2. Section 6-3 verifies the results of the implementation followed by some final conclusions in Section 6-4.

6-1 Model description

6-1-1 Aerodynamic forces

The Breukels [7] model is based on the assumption that the aerodynamics of the kite can be approximated by seeing the kite as an assembly of a finite number of connected two dimensional single membrane airfoils in the spanwise direction as depicted in Figure 6-1, called the *wing sections*. The aerodynamic load for each wing section can be determined separately, depending on their shape, angle of attack and velocity. The airflows around the wing sections influence each other, because of the three dimensional nature of the kite. These (probably significant) effects are neglected at first and afterwards reintroduced by applying some corrections to the calculated aerodynamic forces, depending on the spanwise position of the wing section. The shape deformation of a wing section of the kite is determined by the following parameters, as depicted in Figure 6-2.

- The length of the chord c, measured from leading edge to trailing edge.
- The camber $\kappa = \frac{h}{c}$ is the maximum distance h from the canopy to the chord divided by the length of the chord c.
- The thickness $t = \frac{D}{L}$ is the diameter of the leading edge divided by the chord length c.
- The width W of a wing section.

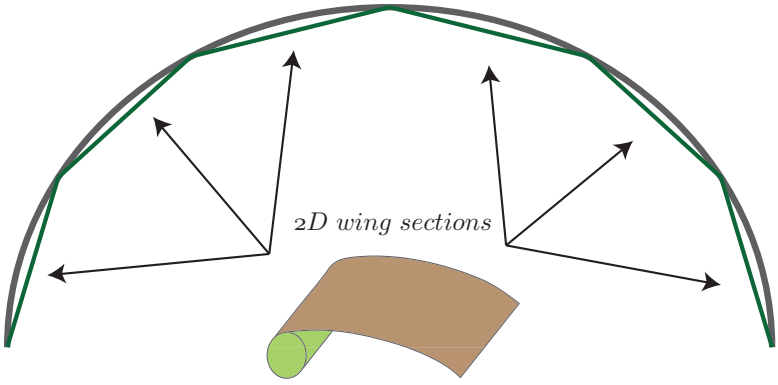


Figure 6-1: The aerodynamic properties of the kite can be determined by the assumption that the kite consists of a finite number of two dimensional single membrane airfoils.

The angle of attack determines the state of the kite.

- The angle of attack α is the angle between the chord and the apparent wind speed in the XZ-plane of a wing section.

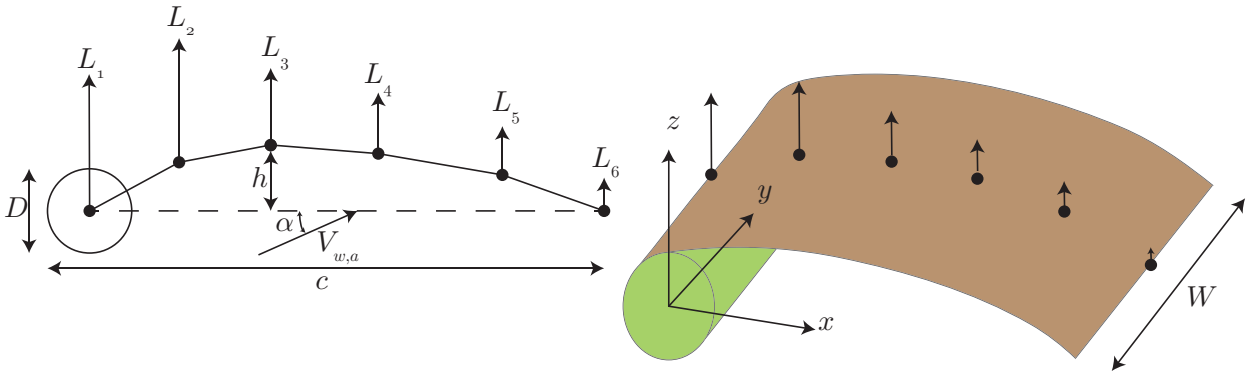


Figure 6-2: The camber $\kappa = h/c$, thickness $t = D/c$, chord length c and angle of attack α of a two dimensional wing section.

These parameters (κ,t,α) were used as inputs to perform a large number of CFD simulations with the program Fluent on a two dimensional airfoil for various values of the inputs. The analysis was performed on three sets of airfoils with a 15%, 20% and 25% thickness. The airfoils within each set ranged in camber from 0% to 12%, and the angle of attack ranged from 0 degrees to 25 degrees. The obtained pressure distributions for each set of parameters were compressed to a lift coefficient C_L, a drag coefficient C_D and an airfoil moment C_M. A fitting procedure resulted in three functions with 60 constants to map $\kappa, t, \alpha \rightarrow C_L, C_D, C_M$. The values of the constants can be found in Breukels [7].

$$C_{L,fit} = \lambda_1(\kappa,t)\alpha^3 + \lambda_2(\kappa,t)\alpha^2 + \lambda_3(\kappa,t)\alpha + \lambda_4(\kappa,t) \tag{6-1}$$

$$C_{D,fit} = \lambda_5(\kappa,t)\alpha^2 + \lambda_6(\kappa,t) \tag{6-2}$$

$$C_{m,fit} = \lambda_7(\kappa,t)\alpha + \lambda_8(\kappa,t) \tag{6-3}$$

$$\tag{6-4}$$

These functions are only valid for $-20\text{deg} < \alpha < 20\text{deg}$. Highly turbulent flows arise for higher angles of attack and the steady CFD analysis does not produce reliable data in that domain. Empirical formulas are used to describe the aerodynamic coefficients C_L and C_D for $-180\text{deg} < \alpha < -20\text{deg}$ and $20\text{deg} < \alpha < 180\text{deg}$, based on Spierenburg [45]. Breukels [7] does not use an empirical formula for the airfoil moment C_M for high angles of attack but uses the fitted function for the whole domain.

$$C_{L,high\alpha} = 2\cos(\alpha)\sin^2(\alpha) \tag{6-5}$$

$$C_{D,high\alpha} = 2\sin^3(\alpha) \tag{6-6}$$

For numerical reasons the formulas for a low and high angle of attack are combined using a smooth transition function σ. The transition takes place around the switching angle α_{sw}.

$$\sigma_1 = \frac{1}{1 + e^{-(\alpha - \alpha_{sw})}} \tag{6-7}$$

$$\sigma_2 = \frac{1}{1 + e^{-(\alpha + \alpha_{sw})}} \tag{6-8}$$

$$C_L = (\sigma_2 - \sigma_1)C_{L,fit} + (1 - \sigma_2 + \sigma_1)C_{L,high\alpha} \tag{6-9}$$

$$C_D = (\sigma_2 - \sigma_1)C_{L,fit} + (1 - \sigma_2 + \sigma_1)C_{L,high\alpha} \tag{6-10}$$

This approach leads to the aerodynamic curves as depicted in Figure 6-3. The curves show some peculiarities. The lift coefficient is relatively high compared to normal aircraft wings and the drag does not always increase with an increasing camber. Furthermore the moment coefficient curve is linear while the CFD results from Breukels [7] show non linear behaviour. Unfortunately the original data is not available to recheck the fitting procedure and these curves have to be used. The lift force, drag force and airfoil moment for each wing segment can be determined by

$$F_L = \frac{1}{2}C_L S \rho_{air} V_{w,a}^2 \tag{6-11}$$

$$F_D = \frac{1}{2}C_D S \rho_{air} V_{w,a}^2 \tag{6-12}$$

$$M = \frac{1}{2}C_M S c \rho_{air} V_{w,a}^2 \tag{6-13}$$

where $S = W_{wingsection} \cdot c$ is the wetted surface determined by the width of the wing section and the length of the airfoil, $V_{w,a}$ the apparent wind speed in the XZ-plane and ρ_{air} the density of the air.

6-1-2 From forces to a distributed load

To distribute the aerodynamic forces over the surface of the kite, each aerodynamic wing section is divided in a finite number of chordwise subsections by n nodes as showed in Figure 6-4. The total lift and drag force will be distributed over these nodes using the constant weight factors w_i while maintaining the same airfoil moment. However, these weight factors are not constant in changing flight conditions, but vary with the angle of attack, camber and thickness. The factors u_i are introduced to include this variation. By introducing the variable a, the total weight factors can be expressed by using one variable a. This idea is illustrated in Figure 6-4.

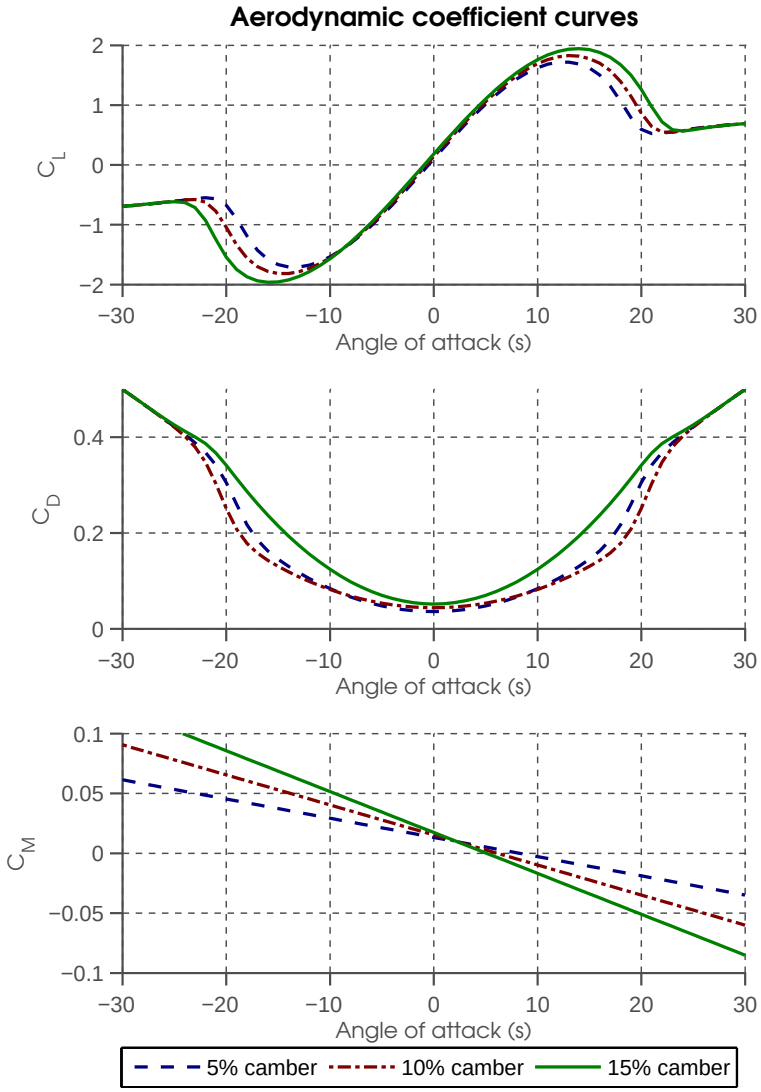


Figure 6-3: The lift coefficient C_L, drag coefficient C_D and airfoil moment coefficient C_M.

The total forces L perpendicular to the chord and D parallel to the chord of a wing section can now be expressed with the two requirements as

$$L = \sum_{i=1}^{n} L(w_i + u_i a) \tag{6-14}$$

$$D = \sum_{i=1}^{n} D(w_i + u_i a) \tag{6-15}$$

$$\sum_{i=1}^{n} w_i = 1 \tag{6-16}$$

$$\sum_{i=1}^{n} u_i = 0 \tag{6-17}$$

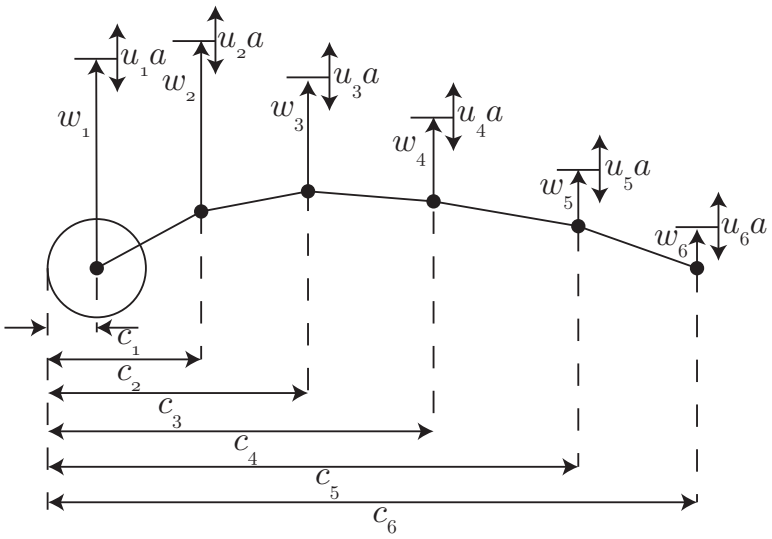


Figure 6-4: A schematic representation of the airfoil model with the distances c_i and weight factors $w_i + u_i a$

A variation of a doesn't change the sum of the aerodynamic forces, but only changes the distribution and therefore only the airfoil moment. The airfoil moment is used to find the parameter a and is taken around the quarter chord point of the airfoil and can be expressed as

$$M = \sum_{i=1}^{n} L(w_i + u_i a)(0.25c - c_i) \tag{6-18}$$

The moments produced by the forces D parallel to the chord are not taken into account, because the force and moment arm of D are small compared to L.

The parameter a can be solved from the previous equations as

$$a = \frac{M - L(0.25c - \sum_{i=1}^{n} w_i c)}{-L \sum_{i=1}^{n} u_i c_i} \tag{6-19}$$

The last step in the procedure is to determine the constant factors w_i and u_i. An infinite number of combinations is possible. The values were obtained by Breukels [7] by comparing the from this procedure resulting aerodynamic forces with the original CFD data. Figure 6-5 shows a comparison between the CFD data and the fitted procedure for an airfoil with 5 nodes, a thickness of 15%, a camber of 4% and an angle of attack of 10.2 degrees.

The values that are used in this thesis are given in Table 6-1.

Table 6-1: Weight coefficients for u_i and w_i.

i	1	2	3	4	5	6
w_i	0.5	0.3	0.1	0.05	0.03	0.02
u_i	1	2	-2	-0.5	0.5	0

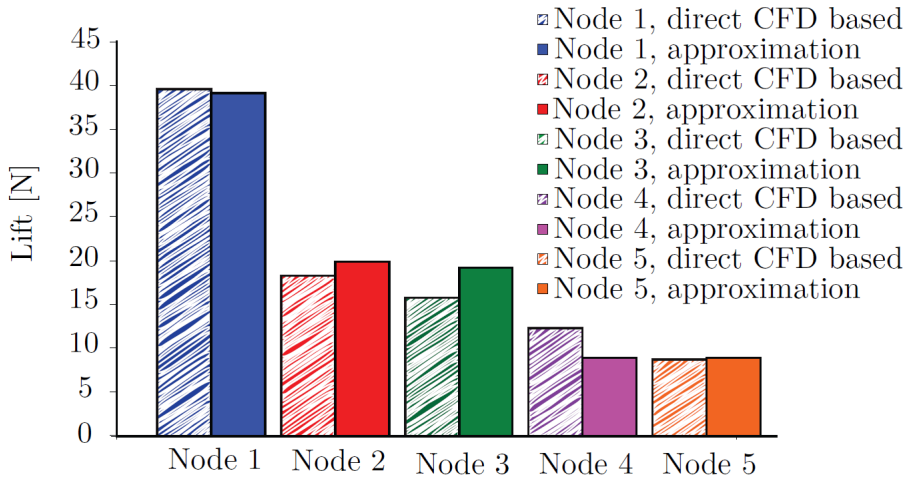


Figure 6-5: A comparison of lift fractions for the approximated and direct CFD based values for an airfoil with a thickness of 15%, a camber of 4% and an angle of attack of 10.2 degrees [7].

6-1-3 Three dimensional correction

So far, this analysis has not included any three dimensional aerodynamic effects that are present in the kite. The aerodynamics of a kite change quite drastically due to effects as span wise flow and tip vortices. Breukels [7] included these effects in his model by analysing the variation of the lift coefficient along the span of the kite with the Tornado vortex-lattice method. The results of this analysis are used to correct the in Section 6-1-1 calculated aerodynamic coefficients for each wing section.

This three dimensional correction factor has not been included in the scope of this thesis. The basic algorithm provides enough information to test the new kite modelling approach. Discussions with the author of the model led to the conclusion that the uncertainty of the three dimensional correction is so significant that it is not clear whether this correction actually improves the model or not.

The end of the tips of the kite don't produce any lift forces, because of the vortexes that occur. Therefore no forces are applied to the end of the tips.

6-2 Implementation in the FE model

The aerodynamic model was specifically developed for the ADAMS kite model [7] and cannot directly be implemented in the finite element model. This section describes how this is done.

6-2-1 Definitions

In chord wise direction, the wing sections are divided in five chord sections, because the weight factors as defined in Table 6-1 are only available for a wing section with six nodes. The original CFD data from [7] is not available to fit weight factors for more nodes. An

interpolation could be made to extend the amount of nodes, but for now this number works quite well with the coarse mesh that was used in the finite element model.

In span wise direction, the wing sections are defined such that they are shifted half the width of the canopy elements relative to those finite elements as depicted in Figure 6-6, so that the mid-line of the wing sections overlaps with a row of finite element nodes. The midpoints of the deflected finite beam elements X_A, X_B in the leading edge are used as the end points of a wing section.

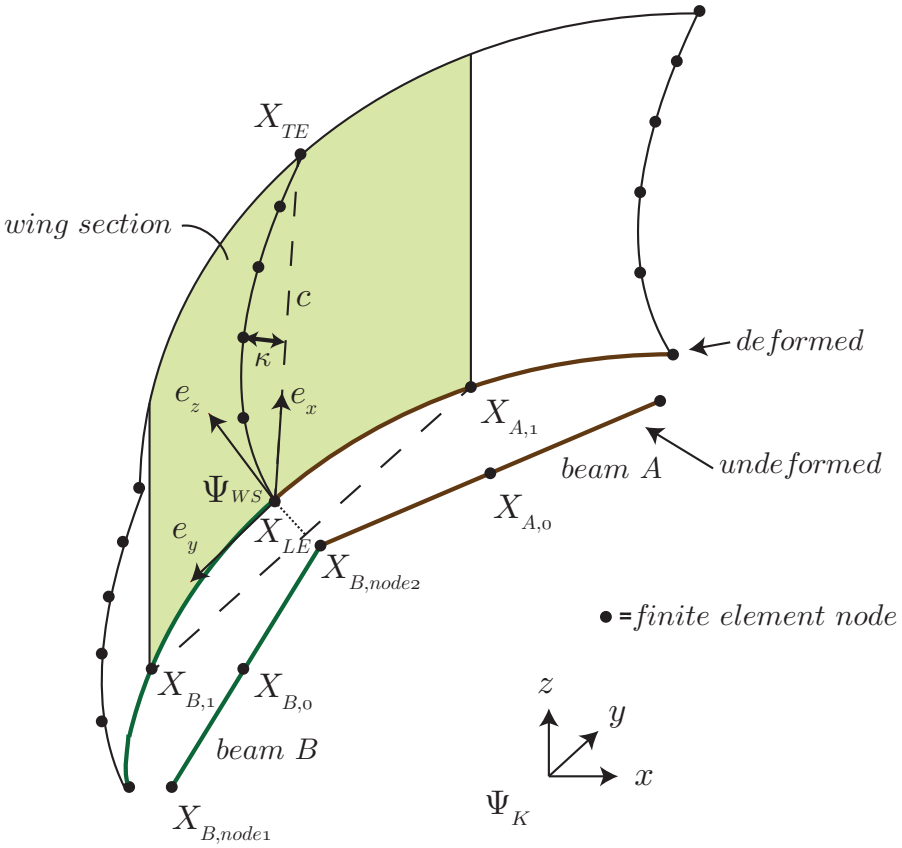


Figure 6-6: The aerodynamic panels are shifted half the width of a canopy element w.r.t the finite elements. The corners of the panel are defined by the mid points of the deformed beam elements X_A, X_B of the leading edge. A straight line between those points is used to build the reference frame Ψ_{WS}.

The mid points of the undeformed finite beam elements i in the leading edge can be calculated from the average of the two beam nodes $\boldsymbol{X}^K_{i,node1}$ and $\boldsymbol{X}^K_{i,node2}$ in the kite coordinate frame Ψ_K.

$$\boldsymbol{X}^K_{i,0} = \frac{\boldsymbol{X}^K_{i,node1} + \boldsymbol{X}^K_{i,node2}}{2} \tag{6-20}$$

The midpoints $X^K_{i,1}$ of the deformed beams i can be calculated from the shape functions using the coordinate $\xi = 0$ that represents the midpoint of the beam, as defined in Appendix A. The vectors $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$ represent the nodal displacements in the local coordinate frame of the beam element Ψ_l as explained in Appendix A. $\boldsymbol{T}^K_l$ is the rotation matrix to rotate the displacements

from the local element frame to the global finite element frame.

$$\boldsymbol{X}_{i,1}^{K} = \boldsymbol{X}_{i,0}^{K} + \boldsymbol{T}_{l}^{K} \begin{bmatrix} \boldsymbol{N}_u(\xi)\boldsymbol{u}^T \\ \boldsymbol{N}_v(\xi)\boldsymbol{v}^T \\ \boldsymbol{N}_w(\xi)\boldsymbol{w}^T \end{bmatrix} \tag{6-21}$$

The wing section coordinate frame Ψ_{WS} is defined in the global finite element kite frame Ψ_K as follows.

- The y-axis points from the midpoint on finite element beam A to the midpoint on finite element beam B.
$$\boldsymbol{e}_{wing}^{y} = \frac{\boldsymbol{X}_{beamB} - \boldsymbol{X}_{beamA}}{\|\boldsymbol{X}_{beamB} - \boldsymbol{X}_{beamA}\|} \tag{6-22}$$
- A temporary vector is constructed from the node at the leading edge to the node at the trailing edge.
$$\boldsymbol{e}_{wing}^{temp} = \frac{\boldsymbol{X}_{TE} - \boldsymbol{X}_{LE}}{\|\boldsymbol{X}_{TE} - \boldsymbol{X}_{LE}\|} \tag{6-23}$$
- The z-axis is defined as perpendicular to the y-axis and temporary axis:
$$\boldsymbol{e}_{wing}^{z} = \boldsymbol{e}_{wing}^{temp} \times \boldsymbol{e}_{wing}^{y} \tag{6-24}$$
- The x-axis is defined as perpendicular to the y-axis and z-axis:
$$\boldsymbol{e}_{wing}^{x} = \boldsymbol{e}_{wing}^{y} \times \boldsymbol{e}_{wing}^{z} \tag{6-25}$$

6-2-2 Aerodynamic parameters for each wing section

The angle of attack α, camber κ and thickness t need to be determined for every wing section to calculate the lift, drag an moment coefficient.

The angle of attack for each wing section is determined from the apparent velocity V_a of the wing section (opposite of the apparent wind speed), and consists of four components.

- The first part comes from the global speed of the kite, expressed by the velocity of the finite element kite frame Ψ_K. This is determined from a translational component $\boldsymbol{V}_k^K$, a rotational component $\boldsymbol{\omega}_k^K$ and the wind speed $\boldsymbol{V}_w^K$.
$$\boldsymbol{V}_{a,global}^{K} = \boldsymbol{V}_k^K - \boldsymbol{V}_w^K + \boldsymbol{\omega}_k^K \times \boldsymbol{X}_{LE}^K \tag{6-26}$$
- The second part comes from the local velocity of the wing section, expressed by the translational and rotational velocity of the frame Ψ_{WS} in the finite element frame Ψ_K. Since the kite is considered to be quasi-static, these velocity terms are not available, because only the static solution to the finite element problem is solved. However from two subsequent time steps n and $n-1$ in the dynamic simulation, the velocity of all the degrees of freedom $\boldsymbol{q}$ can be determined in a numerical way by
$$\boldsymbol{V}_{struct}^{K} = \frac{\tilde{\boldsymbol{q}}_s - \boldsymbol{q}_{n-1}}{h} \tag{6-27}$$

where $\tilde{\boldsymbol{q}}_s$ represent the nodal displacements after structural convergence during the finite element solution process in time step n and $\boldsymbol{q}_{n-1}$ the nodal displacements at the previous dynamic time step and h the time step size. $\boldsymbol{V}^K_{node,i}$ will thus be updated every time when the new aerodynamic forces are calculated during the solution procedure, using the latest known displacements. The local translational velocity of a wing section is now approximated by averaging the speed of the node at the leading edge $\boldsymbol{V}^K_{X_{LE}}$ and trailing edge $\boldsymbol{V}^K_{X_{TE}}$. This term turns out to be quite important for the aerodynamic damping, as will be discussed in Section 6-2-4

$$\boldsymbol{V}^K_{a,local} = \frac{\boldsymbol{V}^K_{X_{TE}} + \boldsymbol{V}^K_{X_{LE}}}{2} \tag{6-28}$$

The local rotational velocity of a wing section can not be taken into account. A local rotation of a wing section results in different angles of attack within one wing section. Since the aerodynamic model can only produce one set of aerodynamic coefficients for a wing section, this effect can not be included.

The total apparent speed of a wing section becomes

$$\boldsymbol{V}^K_a = \boldsymbol{V}^K_{a,global} + \boldsymbol{V}^K_{a,local} \tag{6-29}$$

The angle of attack can be calculated by projecting the apparent velocity on the XZ-plane of the wing section frame Ψ_{WS} by first rotating it with the rotation matrix R^{WS}_K into the wing section frame Ψ_{WS} and then calculating α from the z- and x-components. This is illustrated in Figure 6-7.

$$\boldsymbol{V}^{WS}_a = R^{WS}_K \boldsymbol{V}^K_a \tag{6-30}$$

$$\alpha = atan2(\boldsymbol{V}^{WS}_{a,z}, \boldsymbol{V}^{WS}_{a,x}) \tag{6-31}$$

With this projection, the y-velocity component is neglected. This comes from the two dimensional assumption of the airfoils. This velocity component is low compared to the x and z components and shouldn't not cause any problems.

The length of the chord follows from

$$c = \|\boldsymbol{X}_{TE} - \boldsymbol{X}_{LE}\| + D_{LE}/2 \tag{6-32}$$

with D_{LE} the mean diameter of the leading edge for the wing section.

The thickness is expressed by

$$t = \frac{D_{LE}}{c} \tag{6-33}$$

The camber is calculated by projecting the vectors that point from from all the i nodes within a chord to the leading edge node $\boldsymbol{X}_{LE}$ on the z axis of the wing section frame. The maximum of these values is the parameter h from which the camber is calculated by

$$\kappa = max\left(\frac{\boldsymbol{e}^z_{wing} \cdot (\boldsymbol{X}_i - \boldsymbol{X}_{LE})}{c}\right) \tag{6-34}$$

All the parameters to calculate C_L, C_D and C_M are now defined.

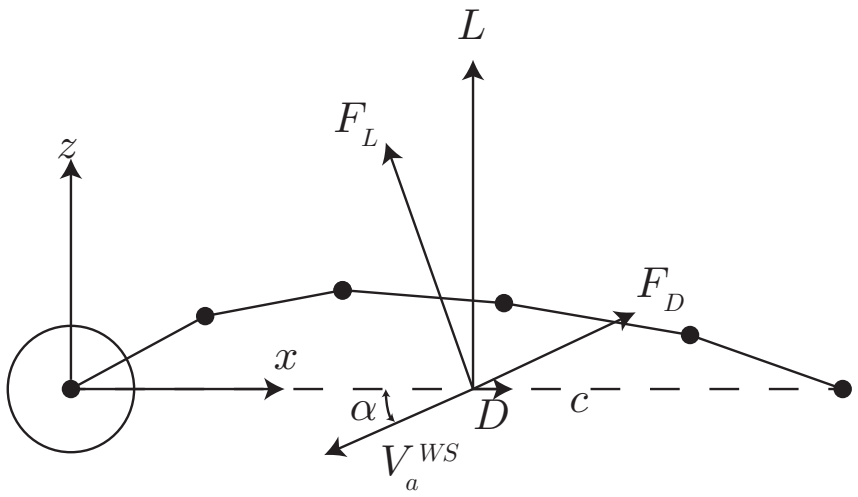


Figure 6-7: The angle of attack α is the angle between the apparent velocity of the wing section $\boldsymbol{V}_a^{WS}$ and the chord c.

6-2-3 Aerodynamic forces

The L and D force for each wing section can be determined from Equation 6-11 and are projected on the x and z-axis of the wing section frame as depicted in Figure 6-7.

$$L = \cos(\alpha)F_L + sin(\alpha)F_D \tag{6-35}$$

$$D = -\sin(\alpha)F_L + cos(\alpha)F_D \tag{6-36}$$

L is used by the in Section 6-1-2 described method to reconstruct a distributed load. The last step is to rotate the forces in the global kite frame Ψ_K so that they can be used in the finite element analysis.

6-2-4 Aerodynamic damping

Both the finite element kite model and the aerodynamic model are quasi-static. The aerodynamic forces only depend on the instantaneous angle of attack α and not on the change in angle of attack $\dot{\alpha}$ in time. When these two are coupled to the dynamic tether model, the finite element kite model will produce a series of quasi-static solutions of the deformation in time. This movement will normally be damping by the aerodynamic drag forces due to a local increased or decreased angle of attack, but these are local velocities in the finite element kite frame and not taken into account due to the quasi-static assumption. This would result in an oscillating kite in the coupled dynamic simulation.

To simulate this aerodynamic damping either damping can be introduced in the dynamic tether model between the bridles attachment points as explained in Chapter 7 or the local speeds of the wing sections need to be approximated numerically to influence the angle of attack as was done in Eq. 6-28.

The latter method is more natural, because it represents what physically happens in the kite. However, it was not possible to create a stable model with only this aerodynamic damping for several reasons. The difference in nodal displacements between two time steps can be quite large in a small time step, especially when something difficult such as buckling of the canopy happens. This creates very high speeds in the direction perpendicular to the canopy, resulting in large angles of attacks. The aerodynamic model did not perform very well for these very high angles of attack and results in instabilities. Furthermore the aerodynamic model was just not designed to include these effects. Oscillations in the kite result in vortices, a lower lift coefficient, different drag forces and other difficult behaviour. The damping effect was included, but only allowed to produce a small amount of damping. Additional damping between the bridle attachment points in the dynamic tether model was required to stabilize the model. A better aerodynamic model is needed to include these effects more realistically.

6-3 Verification

The aerodynamic model was first tested on a simple plate with three chord sections as depicted in Figure 6-8. The used material is the same as the canopy material. Tests were performed with different apparent velocities and directions of the panel. The model performs very well for angles within the range $1deg < \alpha < 20deg$, which is the main flying domain of the kite. Results look, depending on the angle of attack, as depicted in Figure 6-8.

Problems occur and force distributions become less realistic when the angle of attack approaches zero or becomes very large. An example is given in Figure 6-9a where the plate has an angle of attack close to zero and the forces point towards different directions. In reality this is also a complex situation, because fluttering will occur when the angle of attack is zero. Another example is given in Figure 6-9b where the angle of attack is around 50 degrees. The force profile starts to change significantly and is probably not very realistic any more.

The problem lies in the definition of the aerodynamic model. The distributed forces are the result of the chosen weight factors, the C_M coefficient and the parameter a. The model tries to balance the lift forces to obtain a certain airfoil moment. The parameter a in Equation 6-19 can become very large (positive or negative) to obtain a certain airfoil moment and this results in forces that can point in opposite directions. Also the force distribution cannot vary freely, but is predefined to some extent by choosing the u weight factors. Furthermore according to Breukels [7], the calculated moment coefficient is not very reliable and he also experimented with different C_M characteristics in his models, but left it unclear which one was used in his final model. This together, makes the validity of the distribution over the canopy very questionable.

A practical solution to solve the numerical problems that arise is limiting the a-value or changing the weight factors. The latter one is difficult, since the original CFD data is not available and no comparison can be made to see if the new values are realistic. Limiting the a value means that the airfoil doesn't always have the correct airfoil moment, but this will not have a large effect and will therefore be implemented. Furthermore a smooth transition function was added to vary a from zero to the calculated value between $-1deg < \alpha < 1deg$.

The aerodynamic model was tested on the kite for several apparent wind velocities, with the translational degrees of freedom of the four corners fixed. Figure 6-10 shows an example for

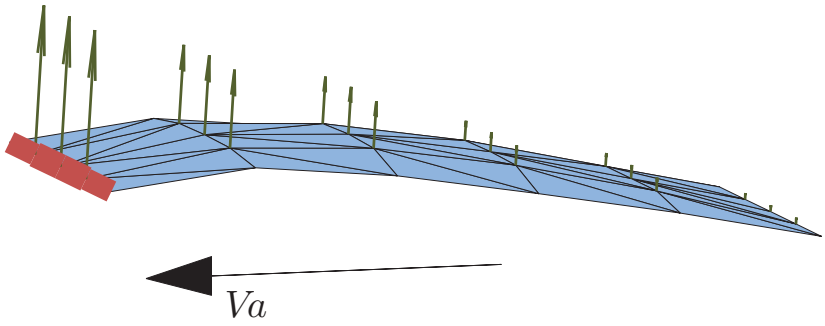
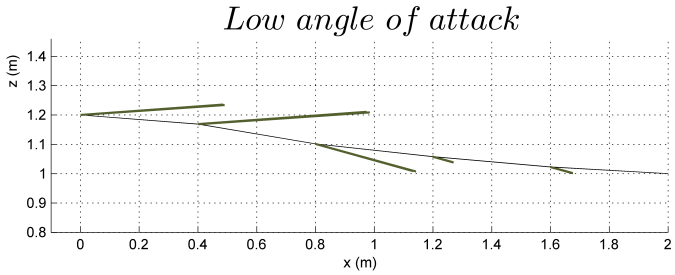
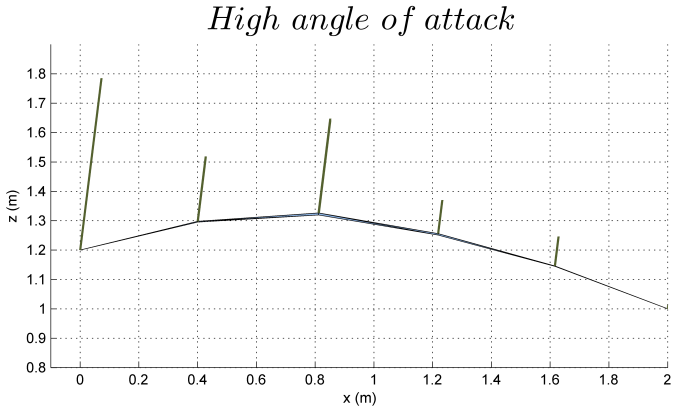


Figure 6-8: An aerodynamic panel with an apparent velocity V_a to test the aerodynamic model. The elements have the same properties as the canopy in the kite.

a wind speed of 30m/s and angle of attack varying from 16 degrees at the tips to 10 degrees in the middle. The several test cases showed that the canopy nicely gets an increased camber and shows realistic behaviour.



(a) The test panel around α=0



(b) The test panel around α=50deg

Figure 6-9: The aerodynamic model gives unrealistic force distributions for very high and very low angles of attack. This is an effect from the method that tries to distribute the lift forces over the chord to obtain a certain airfoil moment.

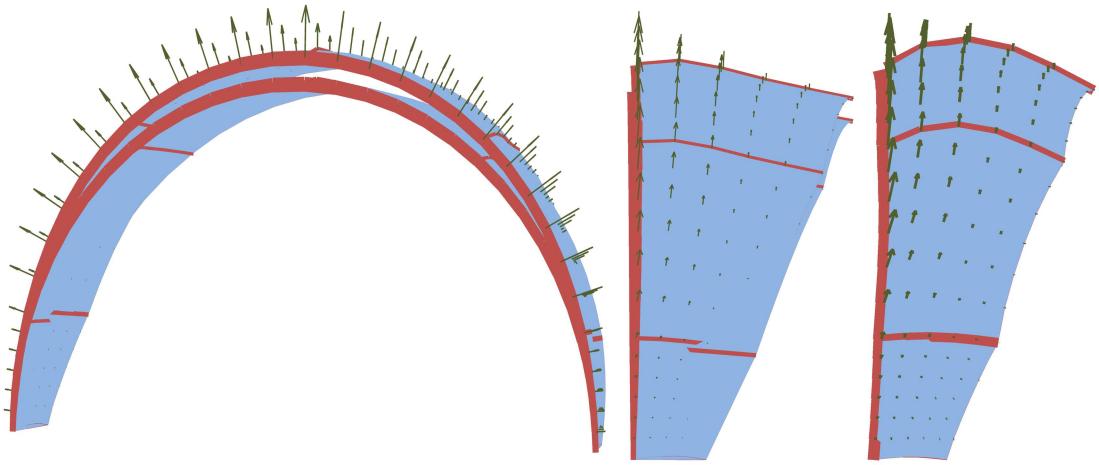


Figure 6-10: An example of the aerodynamic model applied to the kite, hinged at the four corners. Both deformed and undeformed kites are displayed in all figures. The right picture shows the kite with the canopy a factor 10 less stiff to show the effects of the aerodynamic forces.

6-4 Concluding remarks

From the tests it can be concluded that the model suits the purpose of this thesis. It represents the general aerodynamics and returns a distributed load that can be used by the finite element kite model. Some remarks have to be made though.

- The approach taken by Breukels [7] is a bit odd. First very detailed information is obtained from the CFD simulations. This information is then compressed to three aerodynamic coefficients and again expanded to forces on a five nodes. A lot of information is thrown away in this process. It would be better to use the CFD information directly in the approach. This would make it easier to use the model for different mesh sizes of the finite element kite, without having to redetermine all the weight factors.
- The validity of the model is questionable for multiple reasons. The coefficient curves show some peculiarities. The apparent speed is projected on the two dimensional shape of the airfoil and therefore ignoring one velocity component. The parameter a needs to be limited to prevent to model from giving lift forces that work in opposite directions w.r.t the chord, which leads to wrong airfoil moments. The obtained airfoil moment coefficient line is questionable and Breukels [7] uses different ones, but is not clear about that. Furthermore the three dimensional correction factor was not implemented due to the uncertainty whether it improves the model.
- The model returns unrealistic pressure distributions around an angle of attack of $\alpha = 0$ and for very large angle of attacks.
- The model becomes unstable when it is used to generate aerodynamic damping by incorporating local velocities of the nodes in the calculation of the angle of attack.
- The validity of the kite model completely depends on the validity of the aerodynamic model. A better aerodynamic model would therefore give more reliable results of the total simulation. This model acts at the moment as a limitation to the total validity.

Chapter 7

Dynamic tether and bridles model

This chapter describes the dynamic model that simulates the tether and bridles. Section 7-1 explains the modelling choices that were made, followed by the definition of the degrees of freedom and reference fames in Section 7-2. All the forces that act on the system are described in Section 7-3 and the full equations of motion are derived in Section 7-4 with the initial conditions in section 7-5. The description of the inputs for the finite element model that follow from this dynamic simulation are given in Section 7-6. The chapter concludes with some final remarks in Section 7-7.

7-1 Modelling choices

The system to be modelled consists of a tether and four bridles: two power lines connected to the leading edge at points A and C and two steering lines connected to the trailing edge at point B and D as can be seen in Figure 7-1. The four bridles are connected to the control pod that is attached to the end of the tether. The quasi-static finite element kite model exerts forces at the four bridle attachment points. The focus of this thesis is not to develop new cable model, therefore a model is selected that is fast and represents the global behaviour accurate enough. This results in the model as shown in Figure 7-2. This Section discusses the modelling choices that were made.

7-1-1 Bridles

The four bridles need to be able to shorten and lengthen to give the kite the freedom to deform. The forces and lengths of the four tethers change quite significantly when cornering a kite as shown by Breukels [7]. Therefore using rigid elements to model the bridles wouldn't be sufficient and sometimes result in pressure forces on the four tether attachment points. This is not realistic and causes unwanted buckling effects in the canopy that are difficult to solve for the finite element model. The next simplest option is to model them as linear spring-dampers, which allows lengthening and shorting and gives a nice way to control the

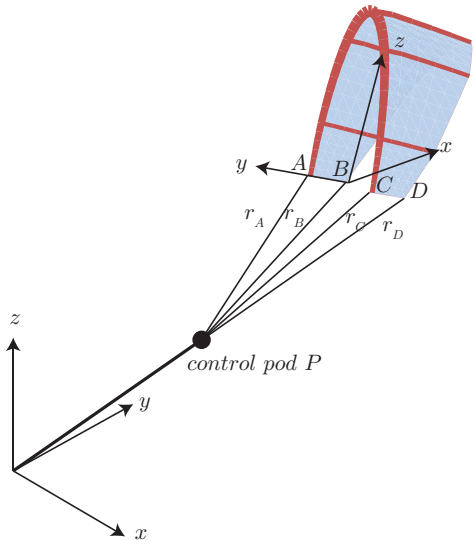


Figure 7-1: Set-up for the dynamic tether and bridles model.

length of the steering lines by shifting their undeformed length. The sag in the bridles can be neglected, because the bridle lines are short, have a low mass and a small diameter resulting in low drag forces. This allows the assumption that the forces in the bridles act in a straight line. Concluding, a massless straight spring will be used to model the bridles.

7-1-2 Tether

The tether is much longer, thicker and heavier than the bridles, resulting in more sag and influence on the kite dynamics. Observations from previous simulations [7, 5] show that the sag can be neglected when flying crosswind up to 100m. Breukels [7] also shows that a kite on a 100m line with a surface in the order of 10 square meters and a tether in the order of 2mm thickness has a ratio between the tether tension and the tether drag of only 0.02 when flying crosswind, which means that the sag of the cable and its weight can be neglected. The sag is also much lower while flying crosswind, than when the kite idles at its zenith position. Since the line length in this research will stay relatively short, the kite is flying crosswind and the main interest is the kite behaviour, the sag of the tether is neglected as well. Flexibility of the tether is also less important to the dynamics of interest and therefore a straight rigid link is used to connect the pod with a tether to the ground.

7-1-3 Bridle attachment points

The kite exerts forces on four masses that are attached to the end points of the bridles to represent the mass of the kite. Most of the dynamic behaviour of the system comes from the aerodynamic drag and lift forces and therefore the exact mass of the bridle points is less important. It is like a spring-damper system with a very high damping constant where most of the dynamic behaviour comes from the damper and the mass of the system is less important. Most of the mass is located at the leading edge side of the kite, therefore the

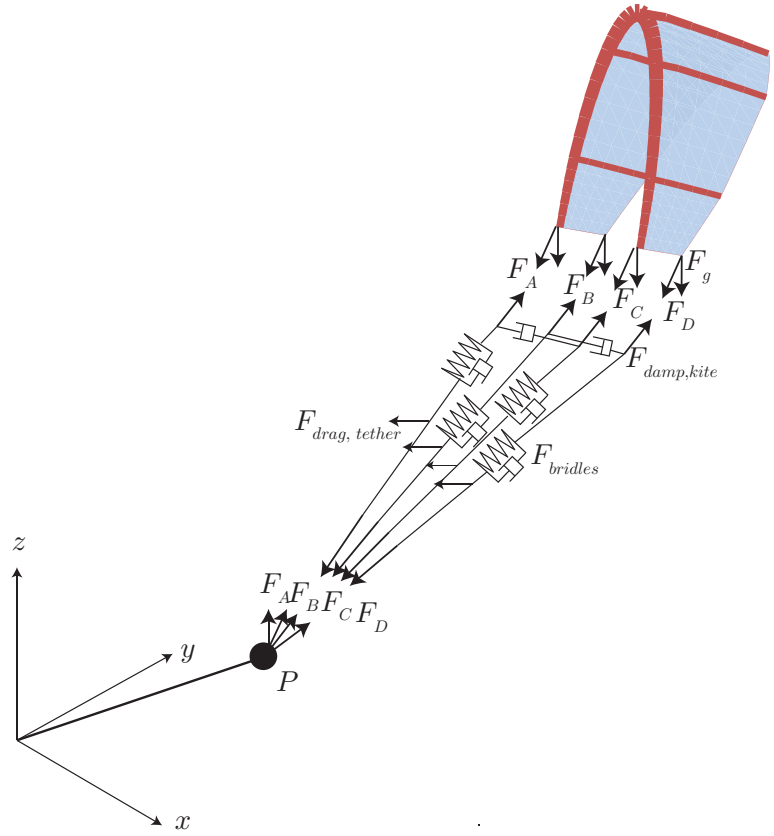


Figure 7-2: Set-up for the dynamic tether and bridles model with the force components.

power lines get 70% and the steering lines 30% of the total mass of the kite, resulting in the masses $m_A = m_C = 3.3kg$ and $m_B = m_D = 1.7kg$.

7-1-4 Connections

In theory these should be all the necessary components to simulate the tether and bridles, but for numerical reasons some additional elements are added. First of all, there is a stiff aluminium bar in the tips of the kite that acts as a very stiff spring between points A-B and C-D. This creates numerical stability problems and forces a time integration algorithm to take very small time steps. This drastically increases the total simulation time, because the fluid-structure-interaction problem needs to be solved for every time step. Therefore a rigid connection is introduced between A-B and C-D in the dynamic simulation. This decreases the flexibility of the kite a little bit, but this is no problem since the displacement in this direction would not be more than a millimetre. An additional effect is introduced, because the aluminium bar in the finite element simulation will bend, but is not allowed to shorten any more and now creates axial forces in the constraint direction. These forces are also applied to the end points of the bridles in the dynamic simulation, but shouldn't cause any problems since they act in the constraint direction and counteract each other. This constraint also decreases the degrees of freedoms, which speeds up the simulation time. Secondly four

dampers between the bridle attachment points are added as depicted in Figure 7-10, to provide some extra numerical damping as was discussed in Chapter 6.

7-2 Coordinates and reference frames

7-2-1 Coordinates

The system has twelve degrees of freedom and can be described by the same number of generalized coordinates. This is preferable above more variables and the use of constraints, since it results in a faster time integration. Furthermore the time integration solvers in MATLAB cannot deal with constraints and it is convenient to be able to use them to easily test the effect of different algorithms. The generalized coordinates y are given by

$$\boldsymbol{y} = \begin{bmatrix} r_A & \alpha_A & \beta_A & \alpha_B & \beta_B & r_C & \alpha_C & \beta_C & \alpha_D & \beta_D & \alpha_P & \beta_P \end{bmatrix} \tag{7-1}$$

where extrinsic Euler angles are used to describe a rotation about the space fixed z-axis with α and a rotation about the space fixed y-axis with β and r_i to describe the length of bridle A and C as shown in Figure 7-3 for the bridle connected to point A [41]. The coordinates of the four bridle attachment points of the kite A, B, C, D and the control pod P can be expressed in these coordinates by

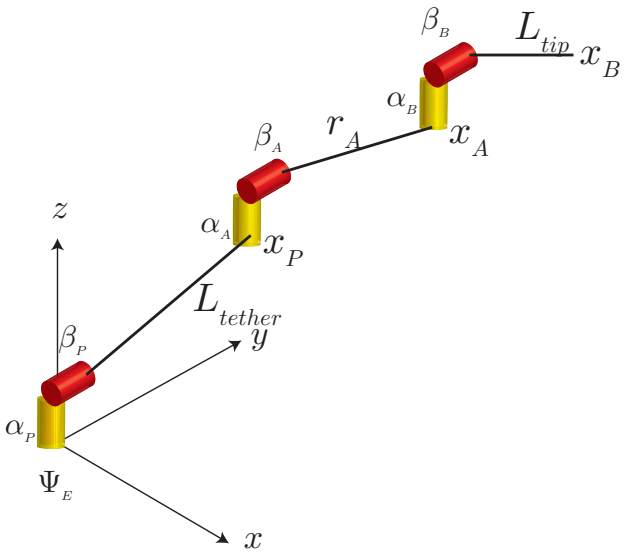


Figure 7-3: Generalized coordinates to describe the position of bridle attachment points A and B and the control pod P. α are the rotations about he fixed z-axis and β the rotations about the fixed y-axis.

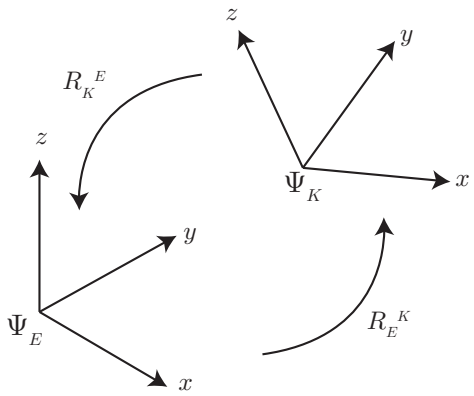


Figure 7-4: The earth reference frame Ψ_E, the floating kite reference frame Ψ_K and rotation matrices R_K^E and R_E^K to convert between them.

$$\boldsymbol{x}_P = R_z(\alpha_P)R_y(\beta_P)\begin{bmatrix}0\\0\\r_P\end{bmatrix} \tag{7-2}$$

$$\boldsymbol{x}_A = \boldsymbol{x}_P + R_z(\alpha_A)R_y(\beta_A)\begin{bmatrix}0\\0\\r_A\end{bmatrix} \tag{7-3}$$

$$\boldsymbol{x}_B = \boldsymbol{x}_A + R_z(\alpha_B)R_y(\beta_B)\begin{bmatrix}0\\0\\L_{tip}\end{bmatrix} \tag{7-4}$$

$$\boldsymbol{x}_C = \boldsymbol{x}_P + R_z(\alpha_C)R_y(\beta_C)\begin{bmatrix}0\\0\\r_C\end{bmatrix} \tag{7-5}$$

$$\boldsymbol{x}_D = \boldsymbol{x}_C + R_z(\alpha_D)R_y(\beta_D)\begin{bmatrix}0\\0\\L_{tip}\end{bmatrix} \tag{7-6}$$

$$\tag{7-7}$$

where R_x, R_y and R_z are the rotation matrices

$$R_x(\gamma) = \begin{bmatrix}1 & 0 & 0\\0 & \cos(\gamma) & -\sin(\gamma)\\0 & \sin(\gamma) & \cos(\gamma)\end{bmatrix} \tag{7-8}$$

$$R_y(\beta) = \begin{bmatrix}\cos(\beta) & 0 & \sin(\beta)\\0 & 1 & 0\\-\sin(\beta) & 0 & \cos(\beta)\end{bmatrix} \tag{7-9}$$

$$R_z(\alpha) = \begin{bmatrix}\cos(\alpha) & -\sin(\alpha) & 0\\\sin(\alpha) & \cos(\alpha) & 0\\0 & 0 & 1\end{bmatrix} \tag{7-10}$$

These coordinates can be combined in the vector $\boldsymbol{x}$ that contains the coordinates of all the masses on which the forces work that are described in the next sections.

$$\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_A^T & \boldsymbol{x}_B^T & \boldsymbol{x}_C^T & \boldsymbol{x}_D^T & \boldsymbol{x}_P^T\end{bmatrix}^T \tag{7-11}$$

7-2-2 Reference frames

The simulation has two important reference frames as depicted in Figure 7-4. The first one is the earth reference frame Ψ_E and the second one the floating kite reference frame Ψ_K. This kite reference frame is attached to the kite and used to communicate between the finite element kite model and the dynamic tether model. All the information that is shared between the two simulations will be represented in Ψ_K: wind speed, kite speed, forces and the displacements of the attachment points of the kite. Vectors represented in the kite reference frame get a superscript K and vectors represented in the earth reference frame the superscript E.

The position and orientation of Ψ_K is recalculated for every time step as follows and as shown in Figure 7-6.

- $\boldsymbol{X}_0^E$ is defined as the mid point between point $\boldsymbol{x}_A^E$ and $\boldsymbol{x}_C^E$ in Ψ_E

$$\boldsymbol{X}_0^E = \frac{\boldsymbol{x}_A^E + \boldsymbol{x}_C^E}{2}$$

- $\boldsymbol{X}_1^E$ is defined as the mid point between point $\boldsymbol{x}_B^E$ and $\boldsymbol{x}_D^E$ in Ψ_E

$$\boldsymbol{X}_1^E = \frac{\boldsymbol{x}_B^E + \boldsymbol{x}_D^E}{2}$$

- The y-axis of Ψ_K becomes

$$\boldsymbol{e}_{y,K}^E = \frac{\boldsymbol{x}_A^E - \boldsymbol{X}_0^E}{\left\|\boldsymbol{x}_A^E - \boldsymbol{X}_0^E\right\|}$$

- A vector that lies in the xy-plane is

$$\boldsymbol{e}_{temp}^E = \frac{\boldsymbol{X}_1^E - \boldsymbol{X}_0^E}{\left\|\boldsymbol{X}_1^E - \boldsymbol{X}_0^E\right\|}$$

- The z-axis of Ψ_K becomes

$$\boldsymbol{e}_{z,K}^E = \frac{\boldsymbol{e}_{temp}^E \times \boldsymbol{e}_{y,K}^E}{\left\|\boldsymbol{e}_{temp}^E \times \boldsymbol{e}_{y,K}^E\right\|}$$

- The x-axis of Ψ_K becomes

$$\boldsymbol{e}_{x,K}^E = \frac{\boldsymbol{e}_{y,K}^E \times \boldsymbol{e}_{z,K}^E}{\left\|\boldsymbol{e}_{y,K}^E \times \boldsymbol{e}_{z,K}^E\right\|}$$

- The unit vectors of Ψ_E are given by $\boldsymbol{e}_{x,E}^E = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$, $\boldsymbol{e}_{y,E}^E = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$ and $\boldsymbol{e}_{z,E}^E = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$

The rotation matrix to rotate vectors from Ψ_K to Ψ_E is given by definition as

$$R_K^E = \begin{bmatrix} \boldsymbol{e}_{x,K}^E \cdot \boldsymbol{e}_{x,E}^E & \boldsymbol{e}_{y,K}^E \cdot \boldsymbol{e}_{x,E}^E & \boldsymbol{e}_{z,K}^E \cdot \boldsymbol{e}_{x,E}^E \\ \boldsymbol{e}_{x,K}^E \cdot \boldsymbol{e}_{y,E}^E & \boldsymbol{e}_{y,K}^E \cdot \boldsymbol{e}_{y,E}^E & \boldsymbol{e}_{z,K}^E \cdot \boldsymbol{e}_{y,E}^E \\ \boldsymbol{e}_{x,K}^E \cdot \boldsymbol{e}_{z,E}^E & \boldsymbol{e}_{y,K}^E \cdot \boldsymbol{e}_{z,E}^E & \boldsymbol{e}_{z,K}^E \cdot \boldsymbol{e}_{z,E}^E \end{bmatrix} \tag{7-12}$$

The used rotation order is yaw(ψ)-pitch(θ)-roll(ϕ) as is shown in Figure 7-5. This leads to the same rotation matrix is in Equation 7-12 from which the three Euler angles can be calculated.

$$R_K^E = R_z(\psi)R_y(\theta)R_x(\phi) \tag{7-13}$$

$$= R_\psi R_\theta R_\phi \tag{7-14}$$

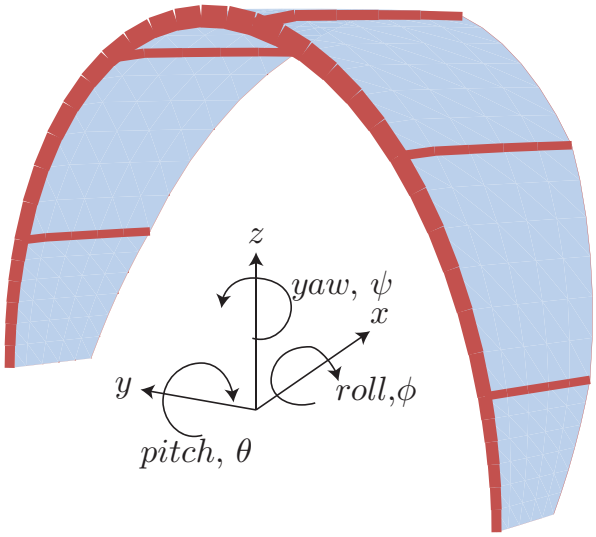


Figure 7-5: Rotations of the kite frame.

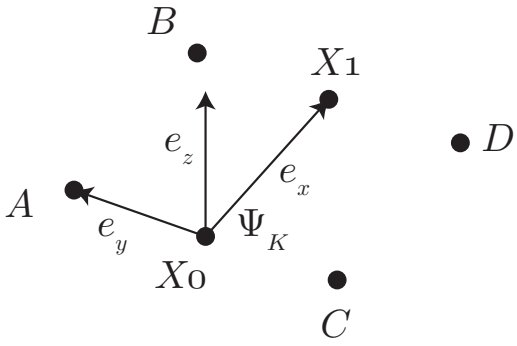


Figure 7-6: The calculation of the kite reference frame Ψ_K from the four bridle attachment points A, B, C, D.

Besides using Euler angles to express the orientation of the kite and compare it to other kite models, four Euler parameters λ_0, λ_1, λ_2, λ_3 are used in the dynamic calculations to avoid singularity problems that arise when using Euler angles as explained by Schwab [42].

$$\lambda_0 = \cos(\mu/2) \quad \boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \sin(\mu/2)\hat{\boldsymbol{h}} \tag{7-15}$$

where μ is a rotation about a certain axis $\hat{\boldsymbol{h}}$.

The rotation matrix is given by

$$R_K^E = \boldsymbol{I} + 2\tilde{\boldsymbol{\lambda}}\tilde{\boldsymbol{\lambda}} + 2\lambda_0\tilde{\boldsymbol{\lambda}} \tag{7-16}$$

with $\tilde{\boldsymbol{\lambda}}$ defined as the skew symmetric matrix

$$\tilde{\boldsymbol{\lambda}} = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \tag{7-17}$$

The Euler parameters can be determined by comparing Equations 7-12 and Equation 7-16.

$$\lambda_0 = \frac{1}{2}\sqrt{R^E_{K11} + R^E_{K22} + R^E_{K33}} \tag{7-18}$$

$$\lambda_1 = \frac{1}{4\lambda_0}(R^E_{K32} - R^E_{K23}) \tag{7-19}$$

$$\lambda_2 = \frac{1}{4\lambda_0}(R^E_{K13} - R^E_{K31}) \tag{7-20}$$

$$\lambda_3 = \frac{1}{4\lambda_0}(R^E_{K21} - R^E_{K12}) \tag{7-21}$$

7-3 Forces

This section describes all the forces that act on the tether and bridles and come from the kite, the flexible behaviour of the tethers, the drag of the tethers, the artificial damping between the tether attachment points of the kite and the gravity.

7-3-1 Kite forces

The finite element kite model returns four force vectors $\boldsymbol{f}^K_{kite,i}$ at the bridle attachment points represented in the kite reference frame Ψ_K that are applied to the four point masses. These can be translated to Ψ_E by

$$\boldsymbol{f}^E_{kite,i} = R^E_K \boldsymbol{f}^K_{kite,i} \quad \text{for } i = \text{A,B,C,D} \tag{7-22}$$

The total force vector that acts on the masses of the dynamic system becomes

$$\boldsymbol{f}^E_{kite} = \begin{bmatrix} \boldsymbol{f}^E_{kite,A}{}^T & \boldsymbol{f}^E_{kite,B}{}^T & \boldsymbol{f}^E_{kite,C}{}^T & \boldsymbol{f}^E_{kite,D}{}^T & \boldsymbol{0}^T \end{bmatrix}^T \tag{7-23}$$

7-3-2 Bridle springs

Figure 7-7 shows a two dimensional perspective of the bridles that are modelled as linear springs.

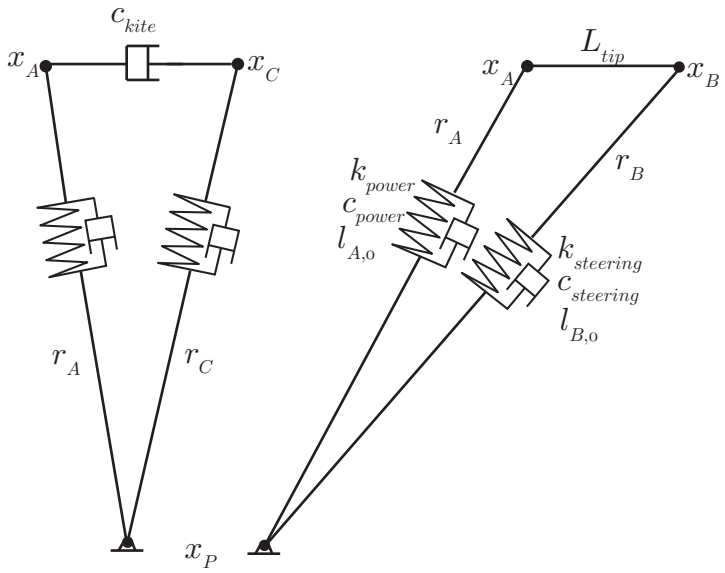


Figure 7-7: Modelling of the bridles connecting the control pod P to the bridle attachment points A, B, C, D in a front view (left) and side view (right) perspective.

The direction vectors of the springs can be expressed by

$$\boldsymbol{e}_i = \frac{\boldsymbol{x}_i - \boldsymbol{x}_P}{\|\boldsymbol{x}_i - \boldsymbol{x}_P\|} \quad \text{for } i = \text{A,B,C,D} \tag{7-24}$$

The speed of the change in length r_i of the springs can be expressed by

$$\dot{r}_i = \frac{\partial r_i}{\partial \boldsymbol{y}} \dot{\boldsymbol{y}} \quad \text{for } i = \text{A,B,C,D} \tag{7-25}$$

where r_i can be calculated by

$$r_i = \|\boldsymbol{x}_i - \boldsymbol{x}_P\| \qquad \text{for i=A,B,C,D} \tag{7-26}$$

$$\tag{7-27}$$

The spring forces that act on the four bridle attachment point masses can be written for all the bridles ($i = A, B, C, D$) as

$$\boldsymbol{f}_{bridle,i} = \begin{cases} (c_{bridle,i}\dot{r}_i + k_{bridle,i}(r_i - r_{i,0}))\boldsymbol{e}_i & \text{for } r_i >= r_{i,0} \\ 0\boldsymbol{e}_i & \text{for } r_i < r_{i,0} \end{cases} \tag{7-28}$$

where $c_{bridle,i}$ is the damping constant and $k_{bridle,i}$ the spring stiffness. The springs can't exert any forces when their length is shorter than de undeformed spring length $r_{i,0}$ since cables are not able to do this. The total force that acts on the control pod is the sum of all the bridle forces working in the opposite direction

$$\boldsymbol{f}_{bridle,P} = -\boldsymbol{f}_{bridle,A} - \boldsymbol{f}_{bridle,B} - \boldsymbol{f}_{bridle,C} - \boldsymbol{f}_{bridle,D} \tag{7-29}$$

The stiffness of the bridles is given by

$$k_{bridle,i} = \frac{EA}{r_i} \tag{7-30}$$

with E the Young's modulus of the cable and A the cross sectional area.

The damping constant is defined by

$$c_{bridle,i} = 2\zeta_{bridle}\sqrt{m_i k_{bridle}} \tag{7-31}$$

where ζ_{bridle} is the relative damping and m_i the mass of the connected bridle attachment point. This shows that short cables are stiffer and have less damping than long cables.

The total force that needs to be applied to the masses becomes

$$\boldsymbol{f}_{bridle} = \begin{bmatrix} \boldsymbol{f}_{bridle,A}{}^T & \boldsymbol{f}_{bridle,B}{}^T & \boldsymbol{f}_{bridle,C}{}^T & \boldsymbol{f}_{bridle,D}{}^T & \boldsymbol{f}_{bridle,P}{}^T \end{bmatrix}^T \tag{7-32}$$

The bridle stiffness $k_{bridle,i}$ will be approximated by making it constant. Small variations in bridle length due to steering movements are in the order of centimetres and will not have a significant effect on the stiffness. The Young's modulus of Dyneema cables that are often used for flying kites is $E_{dyneema} = 100GPa$ and the radius of the bridles that will be used is $r_{bridle} = 0.006m$. Measured damping values of Dyneema show a relative damping of $\zeta_{bridle} = 0.2$, but a value of 0.8 will be used instead. This is higher, but low damping causes oscillations that sometimes result in difficult configurations of the finite element kite model leading to convergence problems, especially during the initialization phase. The bridle properties with a length of 25m bridle are shown in Table 7-1. The force in a power line can go up to 2000N, which would result in a 0.4m elongation of the line which is well within the limits of the Dyneema cable strength that can go up to an elongation of 4%.

Table 7-1: Properties of the bridles

Parameter	Value
k_{bridle}	4.52×10^3 N/m
c_{bridle}	251.7 Ns/m

7-3-3 Tether and bridles drag

Drag forces are exerted on the tether and bridles while sweeping through the air. The longer and thicker the cable, the more significant these forces become. The drag forces are important to create a stable model and they can also be used to add some artificial damping to the model to create a numerically stable system. Drag force on a cable can be defined as [7, 54]

$$f_{drag,cable} = \frac{1}{2} C_D \rho_{air} L d \left\| \boldsymbol{V}_a \right\|^2 \tag{7-33}$$

where $\rho_{air} = 1.225 kg/m^3$ is the air density, C_D the drag coefficient, L the length of the cable, d the diameter of the cable and $\boldsymbol{V}_a$ the apparent wind velocity. The drag force can be split in a tangential and perpendicular component with respect to the cable as can be seen in Figure 7-8. The tangential component has a drag coefficient of $C_D = 0.0017$ and the perpendicular drag coefficient is $C_D = 1.065$ according to Hoerner [20]. Since the tangential part is small compared to the perpendicular part, it is neglected in this simulation.

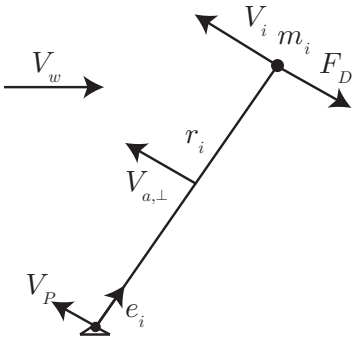


Figure 7-8: Speed and drag forces of a cable.

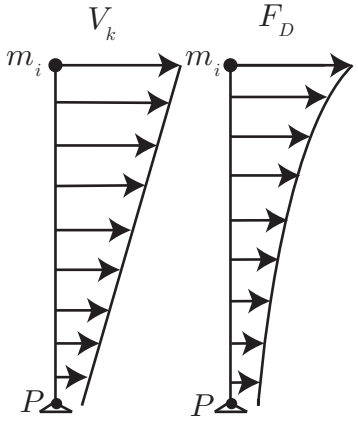


Figure 7-9: Speeds and drag forces along a cable.

The speed along the cable has a linear profile and the drag force increases quadratically with the speed along the length of the cable as is depicted for the bridles in Figure 7-9. For this research it is accurate enough to calculate the resultant drag force from the average apparent speed of the bridle, which is halfway between the mass point m_i and the control pod P as is depicted in Figure 7-8.

$$\boldsymbol{V}_a = \frac{1}{2}(\boldsymbol{V}_i + \boldsymbol{V}_P) - \boldsymbol{V}_w \tag{7-34}$$

The part of this speed perpendicular to the cable is

$$\boldsymbol{V}_{a,\perp} = \boldsymbol{V}_a - (\boldsymbol{V}_a \cdot \boldsymbol{e}_i)\boldsymbol{e}_i \tag{7-35}$$

with $\boldsymbol{e}_i$ the unit vector in the direction from the control P to masspoint i.

The direction vector for the drag force becomes

$$\boldsymbol{e}_{drag,cable} = -\frac{\boldsymbol{V}_{a,\perp}}{\|\boldsymbol{V}_{a,\perp}\|} \tag{7-36}$$

The drag force seen by mass i is now half of the calculated drag force, this results in drag force for a bridle applied at the endpoint of

$$\boldsymbol{f}_{drag,cable,i} = \frac{1}{4} C_D \rho_{air} r_i d \, \|\boldsymbol{V}_{a,\perp}\|^2 \, \boldsymbol{e}_{drag,cable} \quad \text{for } i = \text{A,B,C,D} \tag{7-37}$$

Following the same reasoning the drag force on the tether can be calculated. The total force that needs to be applied to the four masses becomes

$$\boldsymbol{f}_{drag,cable} = \begin{bmatrix} \boldsymbol{f}_{drag,bridle,A}{}^T & \boldsymbol{f}_{drag,bridle,B}{}^T & \boldsymbol{f}_{drag,bridle,C}{}^T & \boldsymbol{f}_{drag,bridle,D}{}^T & \boldsymbol{f}_{drag,tether}{}^T \end{bmatrix}^T \tag{7-38}$$

7-3-4 Bridle attachment points damping

Four dampers are added between points $A-C$, $B-D$, $A-D$ and $B-C$ as depicted in Figure 7-10 to provide some extra numerical damping, because the aerodynamic damping turned out to be unstable as discussed in Chapter 6. The number of dampers is redundant and two cross-dampers would have been enough, but this set-up was chosen to be able to tweak the damping values so that they approximate the aerodynamic damping as good as possible.

The length L_{ij}, speed $\dot{L}_{ij}$ and direction vector $\boldsymbol{e}_{ij}$ can be written for every damper acting between masspoint i and masspoint j as

$$L_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \tag{7-39}$$

$$\dot{L}_{ij} = \frac{\partial L_{ij}}{\partial \boldsymbol{y}} \dot{\boldsymbol{y}} \tag{7-40}$$

$$\boldsymbol{e}_{ij} = \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{L_{ij}} \tag{7-41}$$

The forces on masses i and j become

$$\boldsymbol{f}_{damp,i} = c_{damp,i} \dot{L}_{ij} \boldsymbol{e}_{ij} \tag{7-42}$$

$$\boldsymbol{f}_{damp,j} = -c_{damp,i} \dot{L}_{ij} \boldsymbol{e}_{ij} \tag{7-43}$$

with the damping constant defined as

$$c_{damp,ij} = 2\zeta_{ij}(\boldsymbol{V}_k)\sqrt{m_i k_{kite,ij}} \tag{7-44}$$

where m_i are the mass of a connected bridle point and $k_{kite,ij}$ is the equivalent spring force between the two corner points of the kite and the relative damping

$$\zeta_{ij}(\boldsymbol{V}_k) = \|\boldsymbol{V}_k\| \, \beta \tag{7-45}$$

where β can be used as a parameter to tweak this damping constant from zero damping to an overdamped system. ζ_{ij} is proportional to the speed of the kite $\boldsymbol{V}_k$, because the local aerodynamic damping normally also depends on the kite speed.

The total force that is applied to the masses becomes

$$\boldsymbol{f}_{damp} = \begin{bmatrix} \boldsymbol{f}_{damp,A}{}^T & \boldsymbol{f}_{damp,B}{}^T & \boldsymbol{f}_{damp,C}{}^T & \boldsymbol{f}_{damp,D}{}^T & \boldsymbol{0}^T \end{bmatrix}^T \tag{7-46}$$

To determine the damping values, the kite is seen as if it acts as a spring between between $A-C$ and $B-D$. An experiment with the finite kite model is performed to find these spring constants, while the kite is hinged at the four tether attachment points. First the forces at the four tether points are measured while the kite is loaded with aerodynamic forces due to an apparent flying speed of $V_k^K = [-23 \quad 0 \quad 1.34]^T m/s$. The prescribed displacements (boundary conditions) $\boldsymbol{q}_b$ at the bridle points were taken from a simulation where the kite was flying steady state, which means that the transient movements in the local kite reference frame were damped out. Now extra displacements of $\boldsymbol{q}_{b,C,y} = 0.1$ and $\boldsymbol{q}_{b,D,y} = 0.1$ are applied to points C and D in the y-direction and the resulting force at the tether points is measured again. From this the equivalent spring constants can be determined as

$$k_{AC} = \frac{F_{AC,y}}{u} = \frac{\Delta F_{AC,y}}{0.1} = \frac{19.5N}{0.1m} = 195N/m \tag{7-47}$$

$$k_{BD} = \frac{F_{BD,y}}{u} = \frac{\Delta F_{BD,y}}{0.1} = \frac{5.9N}{0.1m} = 59N/m \tag{7-48}$$

An increase of the displacement by a factor 2 also showed a force increase of approximately a factor 2, which implies that the behaviour can be approximated with a linear spring. It is more difficult to determine the values for the cross-dampers, so the value for β is determined by trial and error using half of the spring stiffness of k_{AC} and the average mass $(m_A + m_B)/2$. $\beta_{AC} = 0.02$, $\beta_{BD} = 0.002$ and $\beta_{AD} = \beta_{BC} = 0.015$ were found to be good values for the relative damping. This allows some oscillations, but makes sure the movement damps out fast enough. The local movements of the kite are also quite well damped in reality as can be seen from test flights. The tweaking of the damping values turned out to be quite important. For example, if the damping of the tether points B and D at trailing edge is too high, those points lacks too much behind in the movement of the leading edge which causes a lot of wrinkling effects in the finite kite model and leads to convergence problems.

7-3-5 Gravity

Gravity force acting on the five mass points is

$$\boldsymbol{f}_g = -9.81\,[0 \quad 0 \quad m_A \quad 0 \quad 0 \quad m_B \quad 0 \quad 0 \quad m_C \quad 0 \quad 0 \quad m_D \quad 0 \quad 0 \quad m_P]^T \tag{7-49}$$

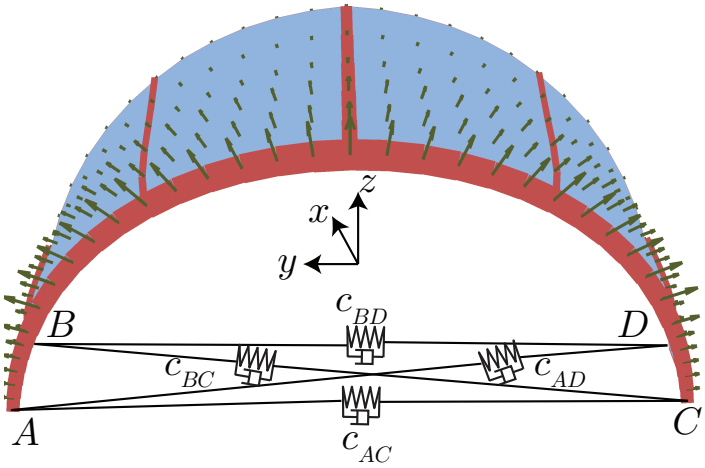


Figure 7-10: Modelling of the damping between the bridle attachment points.

7-4 Equations of motion

The equations of motion are set-up using the information from the preceding sections. The TMT method, as described by Schwab [42], uses a combination of independent generalized coordinates, virtual power and an inertia contribution via d'Alembert forces, a method that can easily be applied here to derive the equations of motion. This section uses Einstein notation instead of vector notation for clarity.

According to Newton

$$\Sigma f_i - M_{ij}\ddot{x}_j = 0 \tag{7-50}$$

Using the virtual power principle, this yields that for virtual velocities holds that

$$\delta\dot{x}_i(\Sigma f_i - M_{ij}\ddot{x}_j) = 0 \tag{7-51}$$

All the coordinates of the centres of masses x_i were expressed before in terms of the independent generalized coordinates y_j in Equation 7-11, this can be expressed by a kinematic transformation T_i

$$x_i = T_i(y_j) \tag{7-52}$$

and the corresponding velocities and virtual velocities become

$$\dot{x}_i = \frac{\partial T_i}{\partial y_k}\dot{y}_k = T_{i,k}\dot{y}_k \tag{7-53}$$

$$\delta\dot{x}_i = T_{i,k}\delta\dot{y}_k \tag{7-54}$$

This result can be substituted in Equation 7-51 and leads to

$$T_{i,k}\delta\dot{y}_k(\Sigma f_i - M_{ij}\ddot{x}_j) = 0 \tag{7-55}$$

Since this must hold for all virtual velocities, we can write

$$T_{i,k}(\Sigma f_i - M_{ij}\ddot{x}_j) = 0 \tag{7-56}$$

The accelerations can be found from differentiation of Equation 7-54

$$\ddot{x}_j = T_{j,l}\ddot{y}_l + T_{j,pq}\dot{y}_p\dot{y}_q \tag{7-57}$$

The last term is called the convective acceleration term g_j

$$g_j(\dot{y}_k, y_k) = T_{j,pq}(y_k)\dot{y}_p\dot{y}_q \tag{7-58}$$

Substituting Equation 7-57 and Equation 7-58 in Equation 7-56 results in the equations of motion

$$T_{i,k}M_{ij}T_{j,l}\ddot{y}_l = T_{i,k}\Sigma f_i + T_{i,k}M_{ij}g_j \tag{7-59}$$

Or in matrix vector notation

$$\bar{\boldsymbol{M}}\ddot{\boldsymbol{y}} = \bar{\boldsymbol{f}} \tag{7-60}$$

with the reduced mass matrix

$$\bar{\boldsymbol{M}} = \boldsymbol{T}^T\boldsymbol{M}\boldsymbol{T} \tag{7-61}$$

the kinematic transfer function that is used to transform between $\dot{\boldsymbol{y}}$ and $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{y}}$ and $\ddot{\boldsymbol{x}}$ and to project forces from the body coordinates on to the generalized coordinates is

$$\boldsymbol{T} = T_{i,j} \tag{7-62}$$

and the reduced force vector

$$\bar{\boldsymbol{f}} = \boldsymbol{T}^T(\Sigma\boldsymbol{f} - \boldsymbol{m}\boldsymbol{g}) \tag{7-63}$$

where $\boldsymbol{g}$ is the acceleration term.

The force vector $\boldsymbol{f}$ consists of all the previous defined forces and becomes

$$\boldsymbol{f} = \boldsymbol{f}_{kite} + \boldsymbol{f}_{bridle} + \boldsymbol{f}_{cable,drag} + \boldsymbol{f}_{damp} + \boldsymbol{f}_g \tag{7-64}$$

The equations of motion were symbolically derived in MATLAB .

7-5 Initial conditions

7-5-1 Wind model

To analyse the behaviour of the modelling approach it is sufficient to keep the wind velocity constant. The wind speed varies in reality quite a lot with the altitude as can be seen in Figure 1-1, but in this modelling study the kite flies on a fixed length tether and will therefore always fly around the same altitude. The wind speed also changes due to wind gusts or wind gaps. These disturbances are also not taken into account, because it is better to study the behaviour of the model first for simple conditions and add these effects in a later phase. The flying domain of a $16m^2$ kite as modelled is roughly a wind speed of 5-10m/s. Therefore as nominal parameter $V_w^E = 5m/s$ will be used.

7-5-2 Initial values of the degrees of freedom

The initial values for the degrees of freedom cannot be chosen arbitrarily. First of all because of the physical dimensions of the kite. If for example point A and C are positioned too far apart, the finite element kite model will not be able to solve the problem or will return unrealistic forces. Secondly because the kite needs to be positioned in a realistic position in the wind window. If not, it will for example stall immediately. And thirdly because the bridles suddenly change in length resulting in a different pitch angle of the kite and oscillations that can be problematic for the finite element model. The equations to calculate the initial values of the generalized coordinates will be described in this section.

Simulations are started from a symmetric position (mirrored in $x = 0$) and the length of the power lines is prescribed. Therefore the following initial values can be assigned

$$\alpha_P = 0 \tag{7-65}$$
$$\alpha_C = -\alpha_A \tag{7-66}$$
$$\alpha_B = -\alpha_C \tag{7-67}$$
$$\beta_P = \beta_A = \beta_C \tag{7-68}$$
$$\beta_B = \beta_D \tag{7-69}$$
$$r_A = r_C = \text{prescribed} \tag{7-70}$$

Four unknown values remain. One of them, β_P is free to choose and determines the initial altitude of the kite. From the finite element kite model the initial distances $A - C$, L_{AC} and $B - D$, L_{BD} are known, resulting in the constraint

$$L_{AC} = \|\boldsymbol{x}_A - \boldsymbol{x}_C\| \tag{7-71}$$

From the prescribed β_A, Eq. 7-71 and Eq. 7-70 the parameters α_A and α_C can be determined.

The remaining unknown parameters β_B and α_B can be found by minimizing the following residual function R

$$R_{BD} = \|\boldsymbol{x}_B - \boldsymbol{x}_D\| - L_{BD,kite} \tag{7-72}$$
$$R_B = \|\boldsymbol{x}_B - \boldsymbol{x}_P\| - r_{B,0} \tag{7-73}$$
$$R = R_{BD}^2 + R_B^2 \tag{7-74}$$

where $r_{B,0}$ can be chosen freely and is used to give the kite an initial pitch angle.

Equation 7-74 can be solved for β_B and α_B by using the symbolic toolbox in MATLAB .

So all the initial parameters can be determined by assuming that the kite starts from a symmetric configuration. β_A can be used to determine the initial altitude of the kite and $r_{B,0}$ can be used to give the kite its initial pitch angle. Initial speeds can be set to zero.

7-5-3 Nominal parameters

All the parameters used in the dynamic tether and bridles simulation are summarized in Table 7-2. Unless stated otherwise, these are the used values in simulations.

Table 7-2: Nominal parameters for the dynamic tether model

Parameter	Value	Description
m_A, m_C	3.5kg	mass of bridle attachment point A and C
m_B, m_C	1.5kg	mass of bridle attachment point B and D
m_P	3kg	mass of the POD
β_{AC}	0.02	relative damping constant between AC
β_{BD}	0.002	relative damping constant between BD
β_{AD}, β_{BC}	0.015	relative damping constant between AD and BC
$V_w^E s$	5m/s	wind speed
$E_{dyneema}$	100GPa	Young's modulus of Dyneema
r_{bridle}	0.0006m	radius bridles
ζ_{bridle}	0.8	relative damping coefficient bridle
L_{tether}	80m	length of the bridles
$L_{bridles}$	25m	length of the bridles
k_{bridle}	4.52×10^3 N/m	bridle stiffness
$c_{bridle} = c_{bridle,C}$	251.7 Ns/m	damping coefficient

7-6 Finite element kite model inputs

The finite element kite model needs certain inputs from this dynamical tether simulation to do its calculations, these are expressed in the kite reference frame Ψ_K.

7-6-1 Kite velocity

The velocity of the kite is the velocity of the kite frame Ψ_K. Since the origin of this frame is defined as the midpoint between point A and B, the velocity can be calculated as

$$\boldsymbol{V}_k^E = \frac{\boldsymbol{V}_A^E + \boldsymbol{V}_C^E}{2} \tag{7-75}$$

$$\boldsymbol{V}_k^K = R_E^K \boldsymbol{V}_k^E \tag{7-76}$$

7-6-2 Wind velocity

The wind velocity is defined as a constant vector in the earth reference frame Ψ_E, in the kite reference frame this becomes

$$\boldsymbol{V}_w^K = R_E^K \boldsymbol{V}_w^E \tag{7-77}$$

7-6-3 Rotational kite velocity

The rotational kite velocity can be calculated from the rate of the Euler parameters. To obtain the Euler parameter rates, the time derivatives of Eq. 7-15 are taken as

$$\dot{\lambda_0} = \frac{\partial \lambda_0}{\partial \boldsymbol{y}} \dot{\boldsymbol{y}} \quad \dot{\lambda_1} = \frac{\partial \lambda_1}{\partial \boldsymbol{y}} \dot{\boldsymbol{y}} \quad \dot{\lambda_2} = \frac{\partial \lambda_2}{\partial \boldsymbol{y}} \dot{\boldsymbol{y}} \quad \dot{\lambda_3} = \frac{\partial \lambda_3}{\partial \boldsymbol{y}} \dot{\boldsymbol{y}} \tag{7-78}$$

Since these expressions are too difficult to derive symbolically a numerical estimate will be used by taking the their difference between time n and $n-1$ divided by the time step h

$$\dot{\lambda_0} = \frac{\lambda_{0n} - \lambda_{0n-1}}{h} \quad \dot{\lambda_1} = \frac{\lambda_{1n} - \lambda_{1n-1}}{h} \quad \dot{\lambda_2} = \frac{\lambda_{2n} - \lambda_{2n-1}}{h} \quad \dot{\lambda_3} = \frac{\lambda_{3n} - \lambda_{3n-1}}{h} \tag{7-79}$$

From these Euler parameter rates, the rotational velocity of the kite frame can be determined by [42]

$$\begin{bmatrix} 0 \\ \boldsymbol{\omega_k^K} \end{bmatrix} = 2 \begin{bmatrix} \lambda_0 & \lambda_1 & \lambda_2 & \lambda_3 \\ -\lambda_1 & \lambda_0 & \lambda_3 & -\lambda_2 \\ -\lambda_2 & -\lambda_3 & \lambda_0 & \lambda_1 \\ -\lambda_3 & \lambda_2 & -\lambda_1 & \lambda_0 \end{bmatrix} \begin{bmatrix} \dot{\lambda_0} \\ \dot{\lambda_1} \\ \dot{\lambda_2} \\ \dot{\lambda_3} \end{bmatrix} \tag{7-80}$$

Euler parameters can suddenly jump between two values, which would result in very high rates. Therefore a filter is applied to filter those outlier values.

7-6-4 Bridle attachment points

The positions of the bridle attachment points $\boldsymbol{x}_i$ have to be determined in the kite reference frame with origin $\boldsymbol{X}_0^E$

$$\boldsymbol{x}_i^K = R_E^K(\boldsymbol{x}_i^E - \boldsymbol{X}_0^E) \quad \text{for } i = \text{A,B,C,D} \tag{7-81}$$

From this the boundary conditions for the finite element model can be calculated by

$$\boldsymbol{q}_b = \begin{bmatrix} \boldsymbol{x}_A^{KT} & \boldsymbol{x}_B^{KT} & \boldsymbol{x}_C^{KT} & \boldsymbol{x}_D^{KT} \end{bmatrix}^T - \boldsymbol{q}_0 \tag{7-82}$$

where $\boldsymbol{q}_0$ are the original positions of the nodes at the four bridle attachment points in the undeformed configuration.

7-7 Concluding remarks

A dynamic model to simulate the tether and bridles with twelve degrees of freedom was developed. The bridles are modelled as linear spring-damper systems and the tether as a rigid rod. 70% of the kite mass was put on the bridle attachment points of the leading edge and 30% on the trailing edge points. A rigid connection between points $A-B$ and $C-D$ was made to remove high frequency content. Sag in the cables and wind variations were neglected. The equations of motion were derived with the TMT method[42].

Correct damping properties are crucial for a smooth working model. Some extra numerical damping was added between the bridle attachment points to represent the aerodynamic damping. The aerodynamic model is not able to provide this damping due to stability issues.

The approach can easily be extended to kites with a more complicated bridle system. The number of degrees of freedom will unavoidable increase, but this will only slightly increase the calculation time, because solving the fluid-structure-interaction problem is the most expensive. The distribution of the mass over more bridle points will even improve the realism of the simulation.

The initialization of the simulation is difficult, because all the component have to find an equilibrium.

Chapter 8

System integration

This chapter describes how the developed components (finite element kite model, aerodynamic model and dynamic tether and bridles model) are integrated in one system that solves the time integration problem. Section 8-1 describes the time integration method, Section 8-2 describes the developed controller and the chapter concludes with some final remarks in Section 8-3.

8-1 Time integration

This section describes the state-equations that need to be solved in Section 8-1-1 and how the fluid-structure-interaction problem is included in these equations in Section 8-1-2, followed by the selection of a time integration algorithm in section 8-1-3. A flow chart gives a full overview of the system in Section 8-1-4 and the section ends with a discussion on some start up problems in Section 8-1-5.

8-1-1 State equations

The dynamic differential equations of the dynamic tether and bridles model were derived in Chapter 7 and resulted in Equation 7-59 which can be written in the form

$$\ddot{\boldsymbol{y}} = f(\boldsymbol{y}, \dot{\boldsymbol{y}}, \boldsymbol{X}, t) \tag{8-1}$$

where $\ddot{\boldsymbol{y}}$ are the accelerations that are a function of the degrees of freedom $\boldsymbol{y}$ that describe the positions of the control pod and the bridle attachment points, $\dot{\boldsymbol{y}}$ the time derivatives, $\boldsymbol{X}$ the state of the system including the kite velocity $\boldsymbol{V}_k$, the rotational kite velocity $\boldsymbol{\omega}_k$, the velocity of the nodes $\boldsymbol{V}_{struct}$ and the wind velocity $\boldsymbol{V}_w$ as described in Chapter 5. The undeformed spring lengths of the left and right steering lines $L_{0,l}$ and $L_{0,r}$ are added to this, because those will be used as control variables as will be discussed in Section 8-2.

Equation 8-1 can be rewritten in the form of a standard system of two first order differential equations by introducing

$$\boldsymbol{u} = \dot{\boldsymbol{y}} \tag{8-2}$$

The resulting system of first order differential equations becomes

$$\begin{bmatrix} \dot{\boldsymbol{y}} \\ \dot{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \\ f(\boldsymbol{y}, \dot{\boldsymbol{y}}, \boldsymbol{X}, t) \end{bmatrix} \tag{8-3}$$

This system of ODE's has to be solved to get the dynamic behaviour of the kite in time.

8-1-2 Including the FSI problem in the dynamic equations

Solving the quasi-static fluid-structure-interaction (FSI) problem results in the forces that act on the bridle attachment points in the dynamic tether and bridles model. These forces depend on $\boldsymbol{y}$ and $\boldsymbol{X}$ and are therefore a part of the dynamic equations that calculate $\ddot{\boldsymbol{y}}$ as can be seen in Figure 8-1(left) and described in Chapter 7.

Every dynamic solving algorithm does multiple function evaluations to calculate the positions in the next time-step $\boldsymbol{y}_{n+1}$. The amount of function evaluations depends strongly on the type of solver that is used. The most computationally intensive part in this modelling approach is solving the FSI problem. It would thus result in a very slow time integration algorithm if the FSI problem needs to be solved for every time the accelerations $\ddot{\boldsymbol{y}}$ are being calculated (every function call of the solving algorithm).

Therefore the assumption is made that the forces that the FSI analysis returns can be kept constant for the period of one time-step. The FSI analysis is removed from the dynamic equations and only performed once for every time step and the resulting forces $\boldsymbol{f}_{FSI}$ used as external input to the dynamic equations by adding them to the state $\boldsymbol{X}$ as depicted in Figure 8-1(right). This will greatly reduce the amount of time needed for the time integration.

8-1-3 Solver selection

This section discusses selection and testing of a time integration algorithm. The MATLAB code of this thesis will be written in a form so that it can be used in the integrated solver algorithms of MATLAB to be able to test with different solvers.The main criteria for the solver are *stability*, *speed* and *accuracy*.

Implicit or explicit

A division in dynamic solver algorithms can be made between implicit or explicit algorithms.

Explicit solvers only need the state of the system at the current time step to calculate the state in the next time step. The number of needed function evaluations to calculate the next time step depends on the algorithm. The time step needs to be small enough to result in a stable simulation.

Implicit solvers need both the state of the system at the current time step and the next time step. This problem needs to be solved in an iterative manner. Implicit solvers have a better stability and accuracy and are therefore often used in stiff problems. Stiff problems show

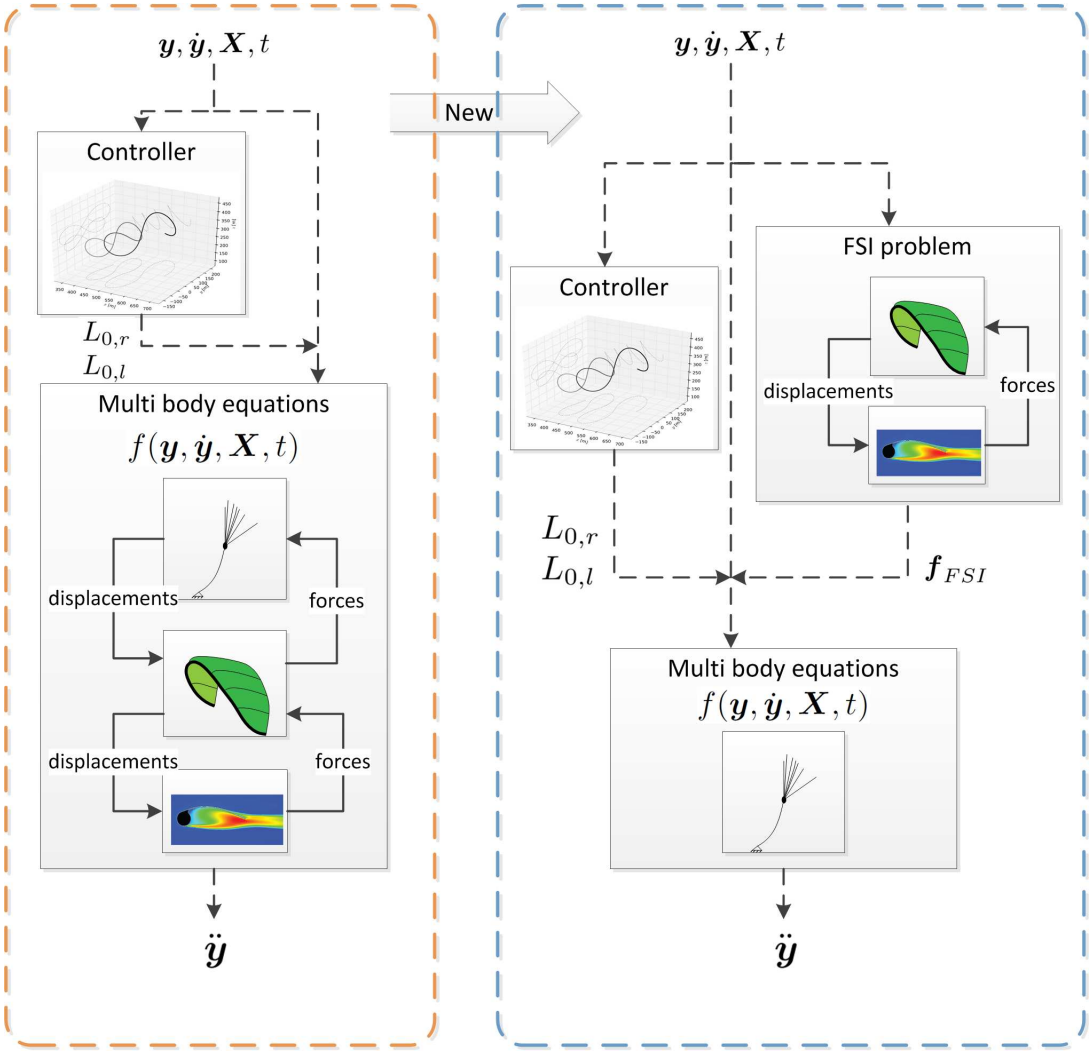


Figure 8-1: Schematic of the function that calculates the accelerations $\ddot{\boldsymbol{y}}$ of the degrees of freedom in the dynamic tether and bridles model. The left shows the original approach with the FSI problem included in the function and the right shows the new approach where the from the FSI analysis resulting forces acts as an input to the function.

high frequency behaviour which would require very small time steps in an explicit method to remain stable, but can be solved with larger time steps using implicit algorithms. The time step can be larger, but the number of function evaluations per time step will be higher. The problem with implicit solvers is that some of them have start-up problems or need gradient information.

The main parameter that influences the speed of the kite simulation will be the time step-size, because the FSI problem only needs to be calculated once for every time step and the function evaluations of the multi body equations are relatively cheap to perform. Therefore implicit solvers are likely to perform better, because they can take a larger time steps. But this also depends on the stiffness of the problem and it is difficult to say which solver will perform better, as will be seen in the next sections.

Stability

The stability and time step of the solver depend on the highest eigenfrequencies in the system. The modelling choices in both the finite element model and dynamic model influence this. There are three sources for high frequency content in this model.

1. The two tip beams in the kite connect both two bridle points (A-B and C-D) and are made from aluminium. These beams act as very stiff springs between the bridle points, resulting in high frequencies in the dynamic simulation. The dynamic equations of this subsystem, depicted in Figure 8-2, can be written as

$$\begin{bmatrix} m_A & 0 \\ 0 & m_B \end{bmatrix} \begin{bmatrix} \ddot{x}_A \\ \ddot{x}_B \end{bmatrix} = \frac{EA}{L} \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_A \\ x_B \end{bmatrix} \tag{8-4}$$

   The eigenfrequencies are determined from

$$det(\boldsymbol{K} - \omega^2 \boldsymbol{M}) = 0 \tag{8-5}$$

   Resulting with the material properties as described in Chapter 5 in an eigenfrequency of $\omega = 2 \times 10^3 rad/s$. This would for instance result in a time step h that should be $h < 1.4 \times 10^{-3} s$ for the often used explicit Runge-Kutta 4 algorithm to be stable, because the RK4 algorithm has the stability criterion $h\omega < 2.8$. To avoid this behaviour a distance constraint between bridle points A-B and C-D was introduced in Chapter 5 to remove this stiff spring. This problem does not occur between the by the leading edge connected points A-C and by the trailing edge connted points B-D. The eigenfrequency resulting from the leading edge that acts as a spring between those points is only $\omega = 10.6 rad/s$ and even smaller for B-D.

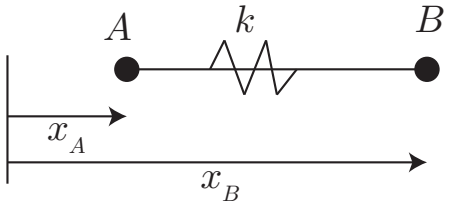


Figure 8-2: The tip beam between bridle point A and B acts as a stiff spring.

2. The bridles in the dynamic simulation are modelled as springs and also relatively stiff. The eigenfrequency can be calculated from the properties given in Chapter 5 by

$$\omega = \sqrt{k/m} = \sqrt{4.52 \times 10^3 / 1.5} = 55 rad/s \tag{8-6}$$

   Resulting in a maximum time step-size $h = 0.056 s$ to get the RK4 method to be stable.

3. The aerodynamics also have their effect on the frequencies in the system. When the forces are very high and change quickly from direction, the small masses to which they are applied accelerate and decelerate very fast. This can be another reason for high frequent behaviour and a small time step. It is difficult to give a good estimation for the frequencies that this will produce.

By removing the high frequency content generated by the kite, it can be concluded that both implicit and explicit methods should result in reasonable time steps. The question is whether an implicit method will still be faster now that the a reasonable time step can be obtained using explicit solvers and what frequencies the aerodynamic forces introduce.

Error

Different numerical integration algorithms lead to different errors. This error originates from two different sources as explained by Schwab [42]. The local truncation error is the error after one time step, these errors add up over the total time span and the order of magnitude depends on the algorithm. Round-off errors come from the fact that computers represent numbers with a finite amount of errors. The global error E is the combination of both. Figure 8-3 shows that there is always an optimal time-step and further decreasing it won't lead to a more accurate solution, because the round-off errors start to increase.

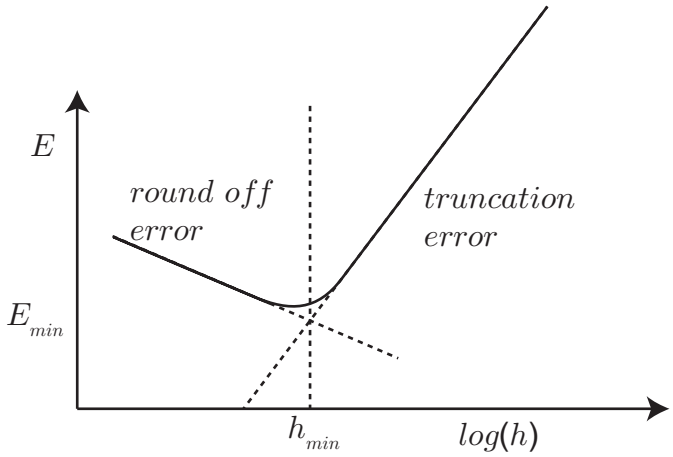


Figure 8-3: The truncation error and round-off error due to numerical integration depending on the time step h.

MATLAB uses two values to define the maximum allowable errors in its algorithms.

- The relative error (RelTol) tolerance is a measure of the error relative to the size of a solution component. It basically controls the number of correct digits in every solution component.
- The absolute error (AbsTol) tolerance is a threshold below which the value of a solution component is unimportant. It determines the accuracy when the solution approaches zero.

The MATLAB solving algorithms approximate the error after every time-step and adjust the step-size so that the error of a solution component fulfils the requirement

$$|e(i))| \leqslant max(RelTol \cdot |y(i))|, AbsTol(i))) \tag{8-7}$$

where $e(i)$ is the error and $y(i)$ the value of solution component i. The used tolerances are RelTol = $2{\times}10^{-3}$, which means that every element in the solution vector is accurate to 0.2% and AbsTol = $1{\times}10^{-6}$, which means that only the first 6 digits are important.

Time stepping

The relation between the size of the time step and the speed of the calculation is not as obvious as it may seem.

1. For every time step, the FSI problem needs to be solved. Since this is an expensive calculation, the time step should be as large as possible.
2. Solving the FSI problem becomes slower when the time step increases. The difference in structural displacements between two time steps will be larger, which makes that the FSI solving algorithm needs more iterations to solve the problem. The aerodynamic forces and stiffness matrix of the kite need to be recalculated for every iteration, making this a very expensive problem.

So increasing the time step has advantages and disadvantages and experiments are needed to find out what the optimal time step is.

Several solving algorithms make use of variable time stepping to find the optimal time step and speed-up the solution process. This approach could really speed up the process. The disadvantage is that controllers work better with fixed time stepping algorithms.

Speed

The speed of the full simulation can be influenced by the following parameters.

- The time step size.
- The tolerances of the time integration algorithm.
- The tolerances for the finite element structural convergence process and the maximum number of allowed iterations.
- The tolerances of the aerodynamic convergence process and the maximum number of allowed iterations.
- Size of the finite element mesh.
- Frequency of changes in the system that come from aggressive steering inputs or wind speed and direction changes.
- How and in what language the code is written.

Experiments with several solvers

Several MATLAB solvers have been tested and compared. A 30 seconds real time simulation was performed where the kite flies a couple figure eight trajectories with a controller, as depicted in Figure 8-9. In the beginning of the trajectory some fast movements are needed to follow the figure eight. The following settings were used:

- Computer: Intel core i5 750 @ 2.8GHz, 4GB memory,
- MATLAB code with some parts compiled to C++.
- Dynamic tether and bridle model tolerances: RelTol = 2×10^{-3}, AbsTol = 1×10^{-6}.
- Structural and aerodynamic tolerances as explained in Chapter 5.

Table 8-1 shows the results. The explicit ODE45 algorithm was by far the fastest, resulting in a speed that is 27.5 times slower than real-time. It can clearly be seen that the time-step is not the most important parameter. The ODE45 solver used an average time-step of 0.0053 seconds, but was almost 3.5 times faster than the ODE113 solver that used an average time-step of 0.0099 seconds. The number of times the stiffness matrix $\boldsymbol{K}_t$ has to be recalculated is the most important factor as was expected. The ODE45 solver will be used in the rest of the simulations.

Table 8-1: Tests with different numerical solvers

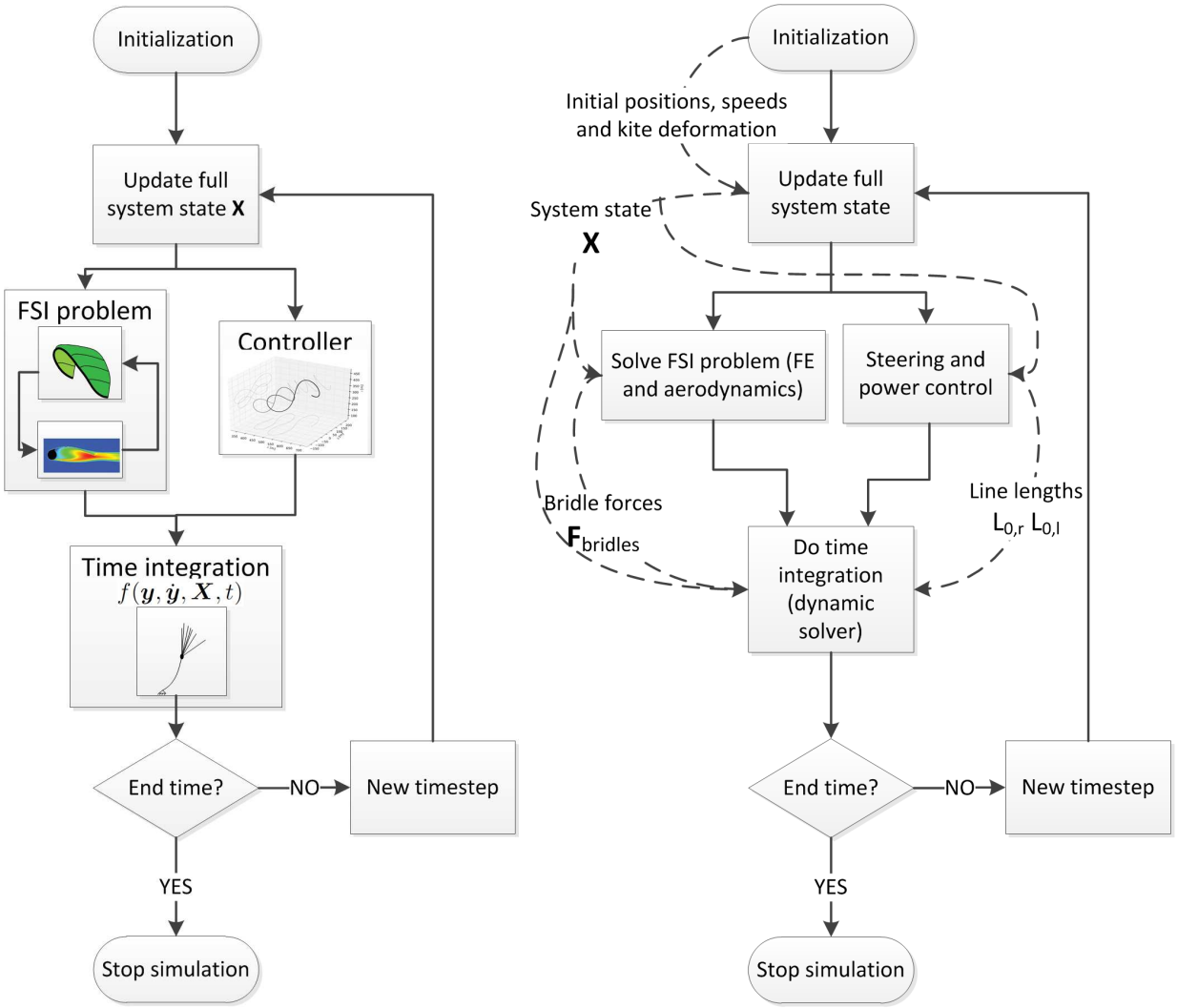
Solver	time (s)	func eval	steps	failed steps	av. h (s)	recalc K_t
ODE45	825	34369	5645	83	0.0053	12237
ODE23	1980	68158	19875	2844	0.0015	27043
ODE113	2724	65202	30585	4031	0.0099	31545
ODE15s	1899	41922	22023	4895	0.0014	22854
ODE23t	1335	27271	12941	2057	0.0023	17497
ODE23tb	3194	99921	37293	1523	0.0008	36004

8-1-4 Flow chart

Figure 8-4a shows the flow chart of the system with all the integrated components. Figure 8-4b shows the same flow chart, but includes the data flows. The several components of the system will be discussed in this section.

During the **initialization** phase all the parameters and initial variable values are set. A selection of the most important ones:

- The finite element model is set up.
- Parameters for the dynamic tether and bridles model: tether length, bridle lengths, damping values.
- The environmental parameters: wind speed.
- Time integration parameters: time step or maximum step size.
- Controller parameters: gains for power and steering control.
- Tolerances for time integration, structural and aerodynamic convergence and maximum number of iterations.
- The initial position and speed of the kite: $\boldsymbol{y}$, $\dot{\boldsymbol{y}}$.
- The initial structural deformation $\boldsymbol{q}$.



(a) Flow chart of the process.

(b) Flow chart of the process including data flow.

Figure 8-4: The time integration process.

During the **full system state update**, all the variables that are needed in the several system parts are calculated and stored. This full state is represented by $\boldsymbol{X}$ and contains the kite velocity $\boldsymbol{V}_k$, the rotational kite velocity $\boldsymbol{\omega}_k$, the velocity of the nodes $\boldsymbol{V}_{struct}$, and the wind velocity $\boldsymbol{V}_w$ as described in Chapter 5. Also all the transformation between different frames are calculated within this step. For example transforming the bridle positions in the global frame to bridle displacements in the kite frame.

The **FSI problem** receives the bridle displacements and the system state $\boldsymbol{X}$ and returns the forces $\boldsymbol{f}_{FSI}$ on the bridle points and adds them to the state $\boldsymbol{X}$.

A steering **controller** adjusts the line lengths of the steering lines $L_{0,r}$ and $L_{0,l}$ and gives them as an input to the dynamic equations via $\boldsymbol{X}$. A power controller adjust the length of the steering lines as well to power and de-power the kite.

A numerical solver does the **time integration** by solving the dynamic equations several times

to calculate the new positions of the pod and bridle attachment points that are represented by $\boldsymbol{y}$.

When the new vector $\boldsymbol{y}$ has been determined, the algorithm proceeds with the next **time step**. This process repeats itself until the end time has been reached.

8-1-5 Simulation start-up

The start up of time integration is difficult. It is quite unnatural to start with an undeformed kite somewhere in the air and suddenly expose it to wind. The following problems occur.

- The finite element model is not yet deformed. It is suddenly exposed to wind and the deformations will change very rapidly in a short time period introducing high speeds. If the damping is not correctly applied, deformations will get too large and the finite element model will fail or very large oscillations will occur.
- The pitch angle of the kite has to be exactly correct. If the angle is too high, the kite will immediately stall and fall out of the air or the leading edge will buckle, because it bends too much backwards. If the pitch is too low, the canopy will start to flutter, resulting in difficult buckling problems.
- The bridles that are modelled as springs are not deformed yet. Their lengths will increase due to the introduced forces. This will happen very fast and some oscillations will occur before it finds its equilibrium position. These oscillations will also change the pitch angle of the kite. Another effect of these oscillations are very high rotational speeds of the kite frame, causing instabilities in the damping effect in the aerodynamic model.

Some experimenting with the line lengths, wind speed and temporarily switching off aerodynamic damping effects and the controllers, has led to a stable start of the system, giving the finite element kite model and bridles some time to find an dynamic equilibrium. This start-up sequence only has to be done once, because other simulations can start from this equilibrium position.

8-2 Control

To be able to see whether this kite modelling approach works, it is necessary to include controllers. The kite would immediately fall out of the sky without a steering controller, because it is un unstable system. The pitch angle of the kite would constantly be either too high or too low without a power controller, causing stall or a total collapse of the kite. This section introduces a steering controller in Section 8-2-1 and a power controller in Section 8-2-2

8-2-1 Steering controller

The steering controller is based on a controller that was designed by Jehle [23] in Simulink. It was rewritten in MATLAB and modified at some points to make it work with the kite model.

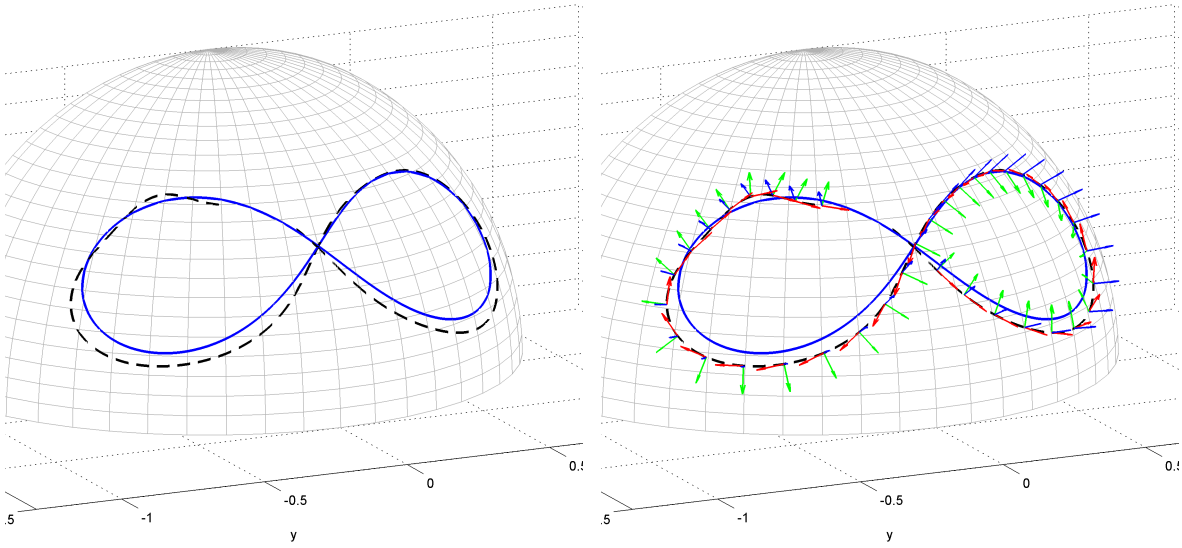
The controller is not optimized or very robust, but was just implemented to be able to fly a trajectory and perform some experiments without having to manually make a list of steering commands to fly a trajectory.

A figure eight on the unit sphere can be described in different ways, one option is to write it in the parametrized form

$$y = \frac{1}{2}\sqrt{2}\frac{\cos(t)}{\sin(t)^2 + 1} \tag{8-8}$$

$$x = \frac{1}{2}\sqrt{2}\frac{\cos(t)\sin(t)}{\sin(t)^2 + 1} \tag{8-9}$$

$$z = \sqrt{1 - x^2 - y^2} \tag{8-10}$$



(a) Desired trajectory (blue, solid) and travelled trajectory (black, dashed)

(b) Kite orientation Ψ_k in flight, blue = z-axis, green = y-axis, red = x-axis.

Figure 8-5: The kite model flying a figure eight.

The kite position can always be projected on this unit sphere by normalizing its position vector. Normally during operation the tether length of the kite continuously changes. This variable is eliminated by projecting its position on the unit sphere. Figure 8-5a shows the desired trajectory and the actual travelled trajectory of the kite. Figure 8-5b shows the kite-frame Ψ_k at certain time intervals on the trajectory.

The kite is able to freely move over the surface of the sphere and therefore has two degrees of freedom left.

The controller uses a different frame Ψ_C to represent the orientation of the kite in than the kite reference frame used by the finite element model. The z-axis is parallel to the tether. The y-axis is perpendicular to the tether and points to bridle attachment point A. The z-axis is perpendicular to both these axis. This is showed in Figure 8-6. The advantage of using this frame representation is that the z-axis of the kite is always perpendicular to the unit-sphere.

Let us define the order of rotations to describe the orientation of the kite by first rotating around the x-axis of the kite, then rotating around the rotated y axis and finally rotating about the rotated z-axis. The last rotation is the yawing of the kite. The big advantage of this rotation order is that the yaw-angle of the kite basically describes the heading of the kite on the unit sphere! This should in theory always point in the same direction as the desired trajectory.

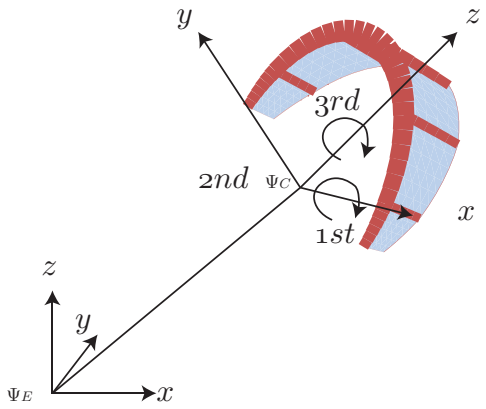


Figure 8-6: The controller frame Ψ_C. Rotation order is defined as XYZ so that the rotation around the z-axis of the kite is always the last one and indicates the heading of the kite on the unit sphere.

The basic controller idea can be described as follows and is illustrated in Figure 8-7.

1. Find the closest point to the kite on the trajectory by solving a minimization problem, described by the parameter $\boldsymbol{s}^*$ in the trajectory equations 8-8

2. Determine the tangential vector $\boldsymbol{t}$ and perpendicular vector $\boldsymbol{p}$ on the unit sphere to the trajectory at this point.

3. The new desired heading for the kite $\boldsymbol{d}_{new}$ is a combination of the vectors $\boldsymbol{t}$ and $\boldsymbol{p}$.

$$\boldsymbol{d}_{new} = \beta \boldsymbol{t} + (1 - \beta)\boldsymbol{p} \tag{8-11}$$

   The parameter β depends on the maximum allowed yawing velocity and the current forward velocity of the kite. When the kite has a low velocity, the kite is able to make a corner with a smaller radius and vice versa. A lower value of β means a sharper corner.

4. The direction of the heading of the kite can be expressed by its yaw-angle α_{kite} on the unit sphere. The direction of the new desired direction $\boldsymbol{d}_{new}$ can be expressed by the angle $\alpha_{desired}$ on the unit sphere.

5. The error between the yaw and desired angle is the error $e = \alpha_{desired} - \alpha_{kite}$ which is the input for the controller.

The exact mathematical expressions can be found in Reference [23]. This basic algorithm is improved in a two ways.

- A simple form of feed forward is introduced by not looking at the point s^* on the trajectory, but looking at the point $s^* + n(V_a)$ that lies a bit further on the trajectory with n a positive number depending on the apparent wind speed V_a. The faster the kite is flying, the further ahead it needs to look. The parameters a_1 and b_1 need to be tuned.

$$n = a_1 + b_1 V_a \tag{8-12}$$

- Also a simple form of gain scheduling is used. The faster the kite is flying, the higher the steering gain $K_{steering}$ needs to be to be able to follow the trajectory. The gain varies linear with the apparent wind speed. The parameters a_2 and b_2 need to be tuned.

$$K_{steering} = a_2 + b_2 V_a \tag{8-13}$$

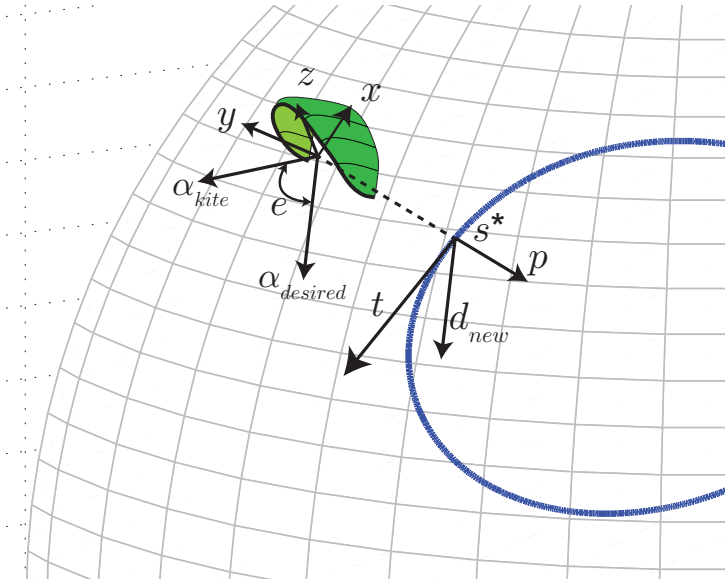


Figure 8-7: The basic working principle of the steering controller.

A simple proportional controller with gain $K_{steering}$ is introduced to control the length difference $\Delta_{objective}$ in the steering lines by comparing the desired heading $\alpha_{objective}$ and actual heading α_{kite}, as displayed in Figure 8-8.

$$\Delta_{objective} = K_{steering}(\alpha_{objective} - \alpha_{kite}) \tag{8-14}$$

Another proportional controller with gain $K_{bridles}$ is used to control the undeformed lengths of the springs $L_{0,r}$ and $L_{0,l}$ that represent the steering lines, which is basically controlling the individual line lengths.

$$L_{0,l} = L_{0,l,init} + K_{bridles}(\Delta_{object} - (L_{0,l} - L_{0,r})) \tag{8-15}$$

$$L_{0,r} = L_{0,r,init} - K_{bridles}(\Delta_{object} - (L_{0,l} - L_{0,r})) \tag{8-16}$$

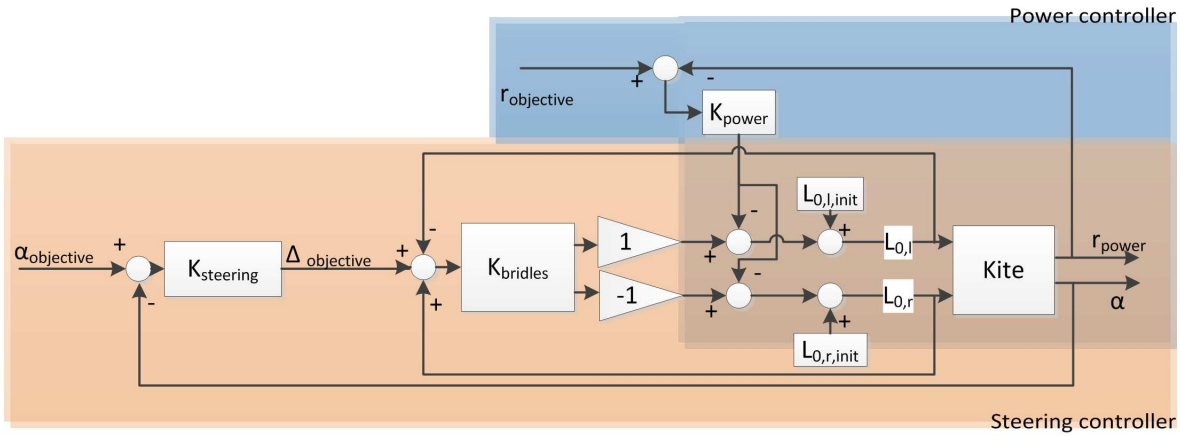


Figure 8-8: The proportional power controller (blue) controls the ratio of force on the steering lines / power lines. The proportional steering controller controls the difference in lengths of the steering lines ($\Delta_{objective}$).

Figure 8-9 shows the result of the working controller. It can be seen that it doesn't follow the trajectory perfectly. One of the properties of a proportional controller is that it will always have an error. Furthermore the controller is not sophisticated enough to deal with the non-linear dynamic behaviour of the kite and the delays in the system. The kite also flies with different speeds in different parts of the trajectory that need different controller gains and tends to slip out of the corner. The used algorithm is too rudimentary to correct for all these effects, but was also never designed to do so. The controller does what it is supposed to do and can be used use in some experiments.

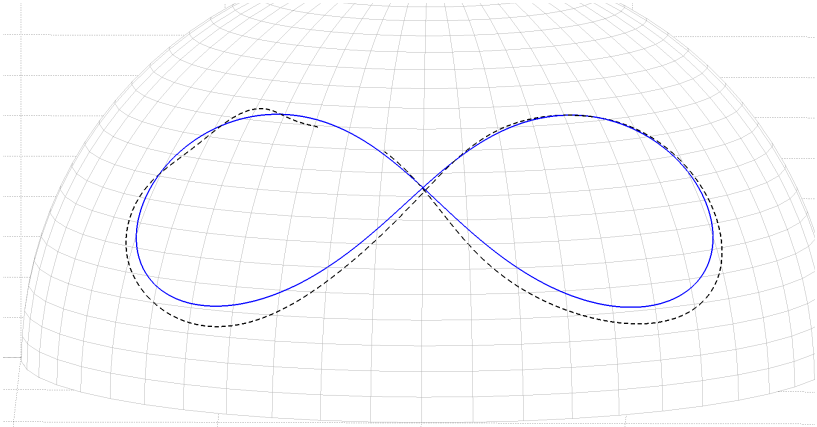


Figure 8-9: The result from a figure eight flight. Blue is the desired trajectory and black dashed the flown trajectory.

8-2-2 Power controller

A second controller is added to control the ratio between the forces on the steering lines and the power lines, because the kite will buckle if the forces on the steering lines become too

high.

$$r_{power} = \frac{F_{steeringlines}}{F_{powerlines}} \tag{8-17}$$

Normally this ratio has a value around 0.3. This controller is not needed in the real system, because the tether will be reeled out faster if the forces become too high. Since the tether length is fixed in this simulation, the forces will become much higher and a controller is needed to keep this ratio around 0.3. A simple proportional controller with a gain K_{power} is used to do this as can be seen in Figure 8-8. The length of both steering lines is controlled simultaneously by changing their undeformed spring lengths L_0 and added to the control input from the steering controller.

$$L_0 = L_{0,init} - K_{power}(r_{objective} - r) \tag{8-18}$$

8-3 Concluding remarks

This chapter described an efficient way to couple the finite element kite model, aerodynamic model, dynamic tether model and a controller together. The fluid-structure-interaction problem was removed from the dynamic equations so that it only has to be solved once per time step, resulting in lower simulation times.

A controller was designed to steer the kite and control its power. The controller is not very robust, but can be used to fly some test trajectories with the system.

Tests with different time integration algorithms show that the explicit Runge-Kutta algorithm (ODE45) with a variable time step is the fastest, resulting in a simulation speed that is 27.5 times slower that real-time. This result is quite good for such a complex system programmed in non compiled code.

Starting the simulations is a difficult phase in the time integration process, because all the components need to find a equilibrium together.

Chapter 9

Results & Discussion

This chapter shows and discusses the results of the introduced modelling approach to see whether it leads to realistic and fast result. The individual components of the model were already partly validated in the previous chapters.

Experiments to fully validate this model are outside the scope of this thesis, because the primary goal of this thesis is to introduce a new modelling approach. A complete validation would be the next step and is a difficult task. None of the currently available models has been fully validated, because the lack of available test data. More test data has become available over the past year, but validation methods are still in development. Comparing measurement data and simulations is not straightforward, because for instance it is difficult to measure the deformations of the kite or to reproduce the the exact measurements conditions in the computer model.

Without experiments it is still possible to say something about the validity of the model by comparing it to other models and comparing the results to known facts about kite behaviour. This approach will be used in this chapter.

This thesis doesn't discuss the energy production part of the kite energy system, but focusses on the modelling of the kite. Therefore the energy generation is not a part of the results won't be discussed.

Section 9-1 describes the used set-up to obtain the simulation data, followed by studies on the steering behaviour in Section 9-2, the aerodynamic behaviour in Section 9-3 and the structural behaviour in Section 9-4. Section 9-5 discusses the results of the time integration algorithm and the chapter concludes with some remarks in Section 9-6.

9-1 Test flight environment

A graphical user interface was designed in MATLAB as shown in Figure 9-1 to perform several test flights and study the travelled trajectory and the deformations modes of the kite.

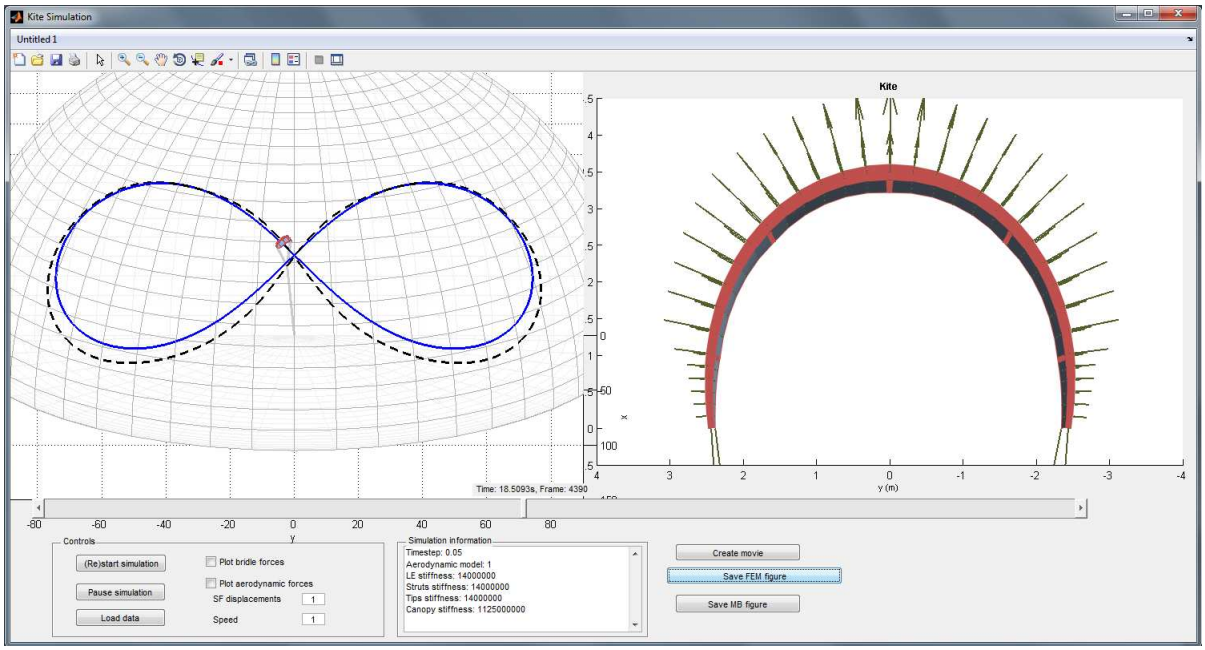


Figure 9-1: The Matlab user interface to analyse the results of test flight trajectories.

One representative figure eight flight was selected for analysis and is depicted in Figure 9-2 where the solid blue line shows the desired trajectory and the dashed black line the travelled trajectory. The kite starts in the middle at point p_0 and travels from p_0 via p_1, p_2 and p_3 to p_4. It takes the kite approximately 20 seconds to fly this trajectory with a constant wind speed of 5 m/s and a 100 m tether length.

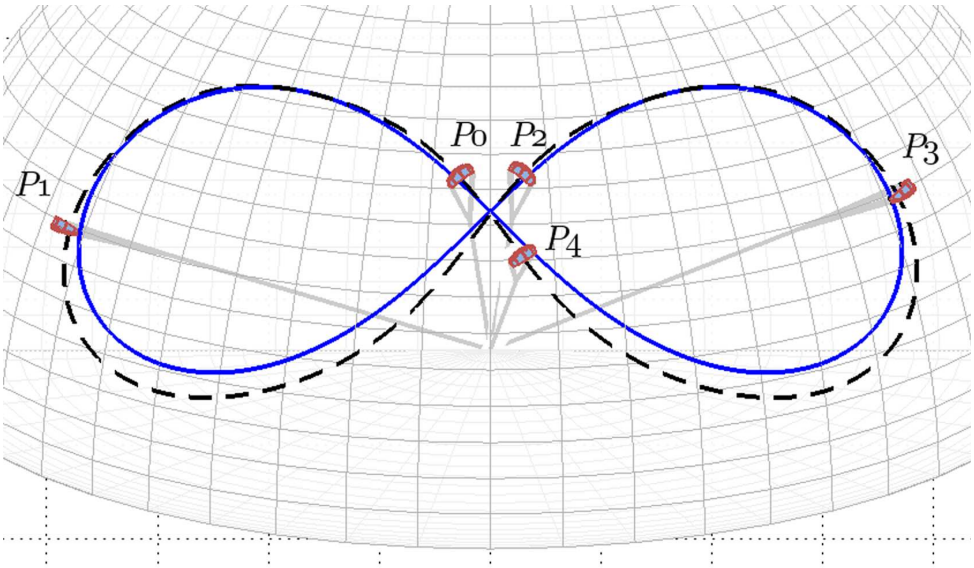


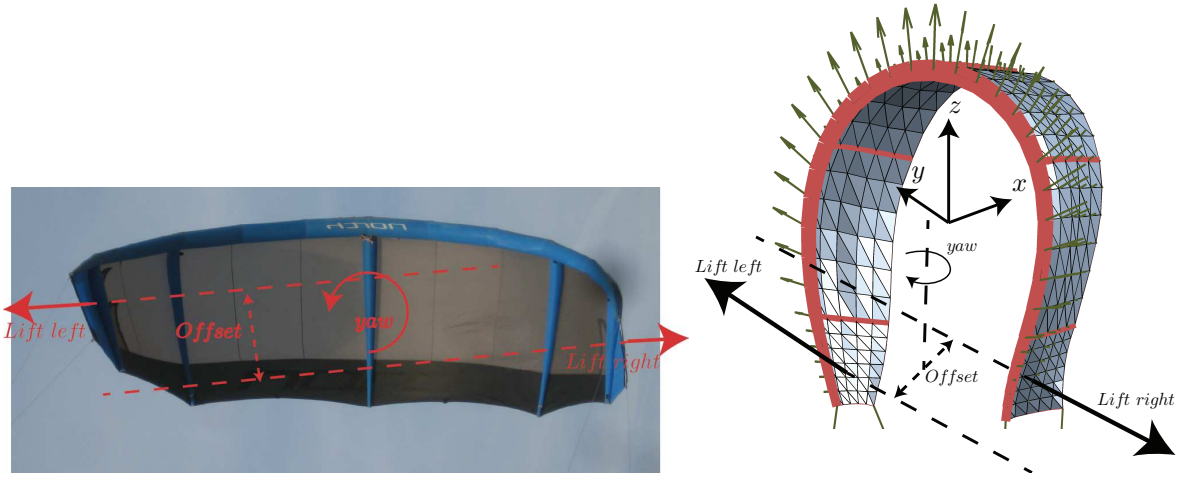
Figure 9-2: The kite flies the selected figure eight trajectory in approximately 20 seconds. The blue line shows the desired trajectory and the dotted line the actual trajectory.

9-2 Steering behaviour

The main reason to model the full flexible behaviour of the kite was to obtain realistic steering behaviour. The cornering behaviour of a kite has been subject of debate for a long time. Kite designers want to develop high manoeuvrable kites for professional kite surfers and more stable kites for beginners. Kites in energy generating systems also need good steering properties to fly a figure eight. Breukels [7] was the first to answer the question how a kite actually corners.

A left corner is defined as the movement resulting from a shortening of the left steering line and vice versa. Starting from the zenith position, this would look as a movement to the left from the perspective of a kite surfer controlling the kite. This looks as a movement to the right from the outer perspective taken in Figure 9-2.

When a kite is steered to the left by increasing the force on the left steering line, the angle of attack at the left tip increases and of the right tip decreases. This results in a higher drag force on the left tip. Kite designers thought for a long time that this was the main reason for the cornering ability of a kite. However, this difference in drag is so small, that it only results in a very slow yaw rate of the kite and can't be the main reason for the cornering ability. The lift force on the left tip also increases compared to the right tip due to the increased angle of attack. But this is no reason for the kite to yaw, only to move sideways to the left. The most important reason for the cornering behaviour is the asymmetric deformation of the kite. The higher force on the left steering line causes the left tip to bend forward towards the leading edge and the right tip to bend backward as shown in Figure 9-3. The deformation of the kite results in an offset between the lift forces on both tips, creating a yaw moment which makes the kite turn/yaw. This hypothesis was confirmed by Breukels [7] by comparing the flexible kite to a rigid kite, showing that the yawing rate of a normal kite is much larger compared to the rigid kite. These conclusions give a basis to verify the cornering properties of the developed kite model in this section.



(a) Bottom view of the real kite.

(b) Model with magnified displacements.

Figure 9-3: Cornering of a kite is a result of the shape deformation that creates an offset in the working lines of the lift forces at the tips. This results in a yawing motion to the left.

9-2-1 Steering input

Figure 9-4 shows how the steering input from the controller results in a line length difference between the steering lines. There is some delay, coming from the spring like behaviour of the lines and the applied delay in the controller. The reason that both steering line lengths change while making a corner is that the power controller also uses these line lengths to control the total power in the kite. Figure 9-5 shows how the steering input relates to the yaw rate of the kite. A line length difference of the steering lines results in a yawing motion in the correct direction, as was expected. There is clearly a relation between yaw rate and controller input. This proportional relationship between the yaw rate of the kite and the steering input was found before by Erhard and Strauch [13] and Jehle [23] from different experimental datasets. That the developed model shows the same proportional relationship as from experimental data is a first confirmation of the validity of the model.

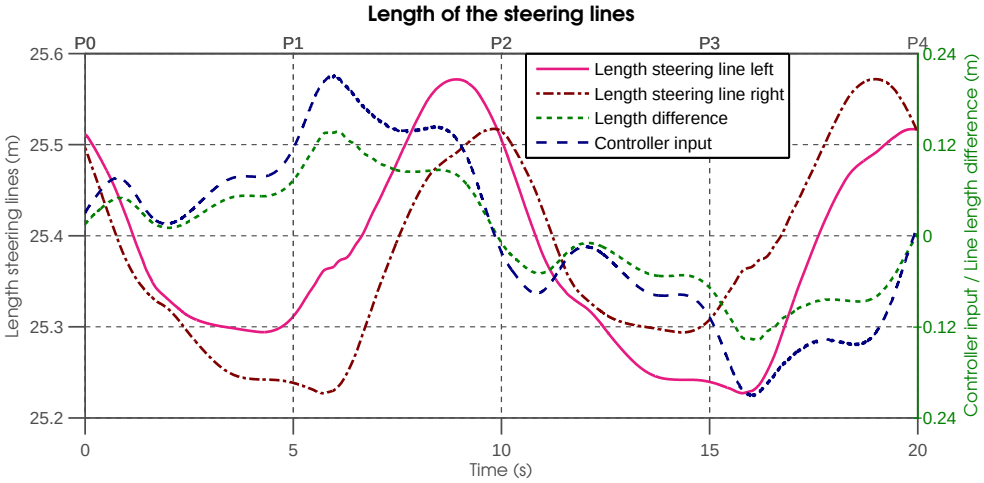


Figure 9-4: Length of the steering lines while flying a figure eight.

Figure 9-5: Relation between the steering input and yaw rate while flying a figure eight.

9-2-2 The steering deformation of the kite

Figure 9-6 shows the deformation of the kite while cornering. The asymmetric deformation can clearly be seen. While cornering to the right, the right tip bends forward and the angle of attack increases while the left tip bends backward and the angle of attack decreases and vice versa. This creates the resulting offset in lift forces resulting in yawing of the kite. This looks very natural compared to Figure 9-3 and is according to the description of Breukels [7].

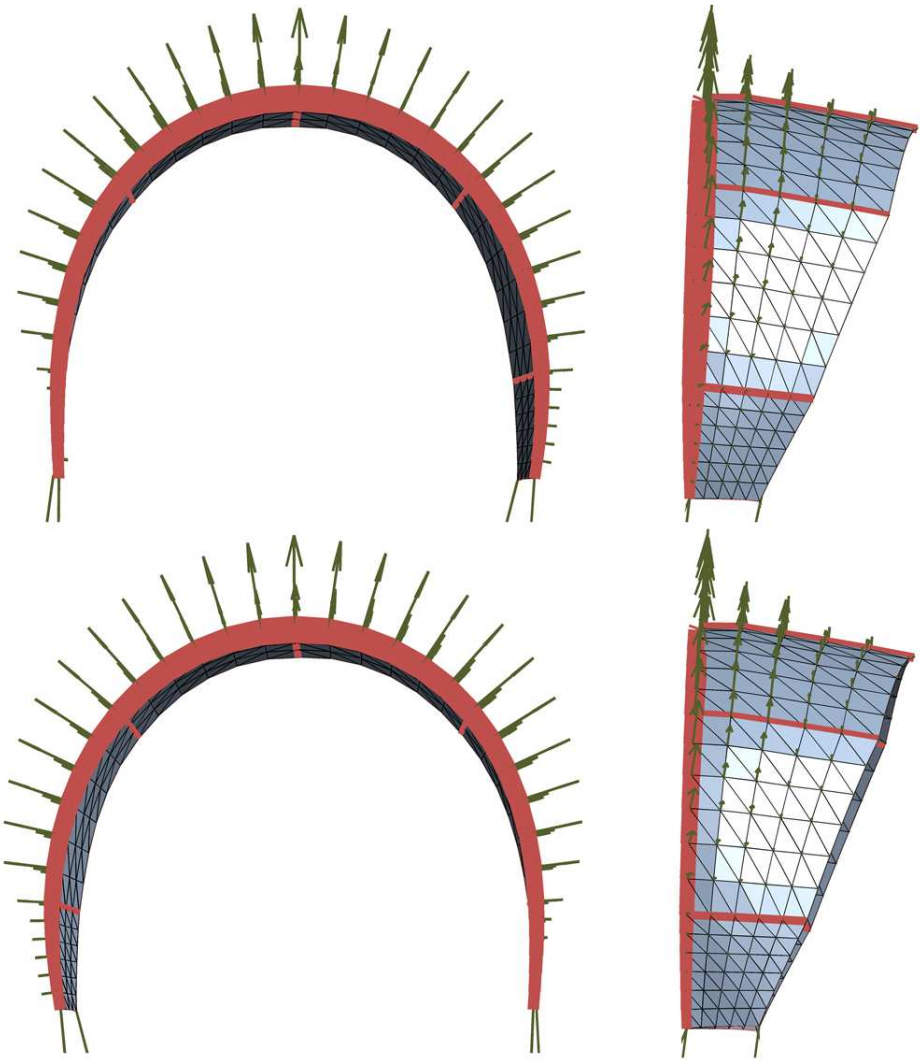


Figure 9-6: Deformation of the kite while cornering to the right (top) and left (bottom).

9-2-3 Angle of attack

Figure 9-8 shows that the angle of attack increases on the right tip and decreases on the left tip while cornering to the right and vice versa. The angle of attack at the left tip even becomes negative for a while. The higher angle of attack results in a higher lift force and drag force at the right tip, as can be seen from Figure 9-9. The drag force is only slightly

higher and has a minor result on the yaw rate compared to the lift force. This is exactly what was expected and described by Breukels [7]. Figure 9-7 shows that the angle of attack gradually varies over the kite from tip to tip. The angle of attack of the middle wing section stays almost constant and results can be mirrored from left to right around the time of 10 seconds. Some oscillations can be seen at the tip sections right after passing point p_1. This is the result of a fast increase in the steering input to be able to follow the trajectory. The angle of attack of the left tip becomes negative here. Switching from a positive to a negative angle of attack, causes the sign of the lift force to flip. This results in some oscillations. From movies of the real system similar oscillations have been seen at the tips.

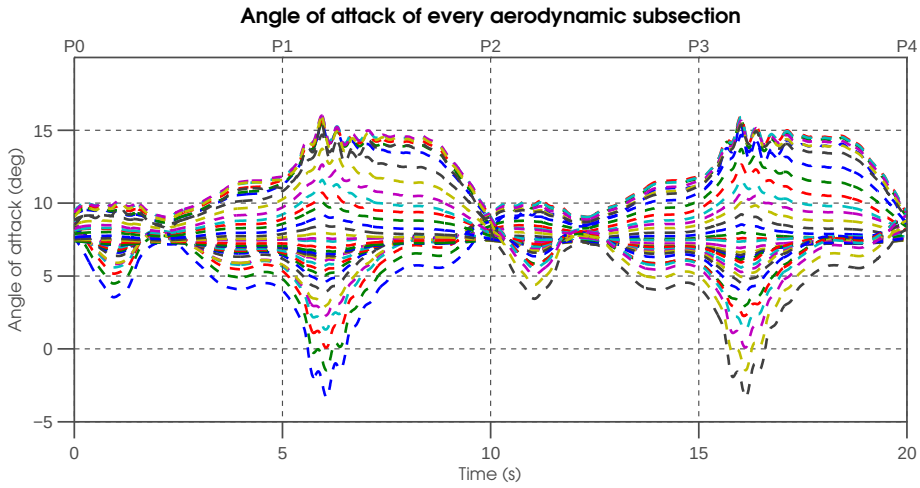


Figure 9-7: The angles of attack of all the subsections of the kite while flying a figure eight.

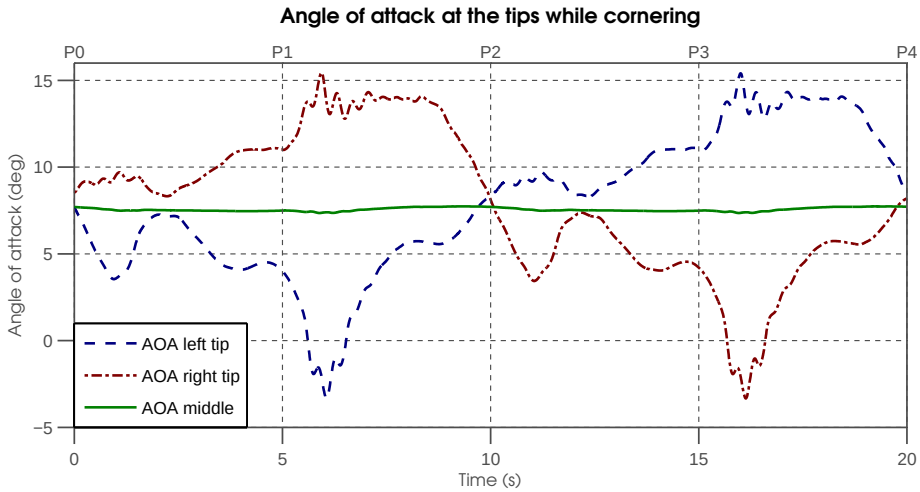


Figure 9-8: The angles of attack of the left tip, right tip and middle section of the kite while flying a figure eight.

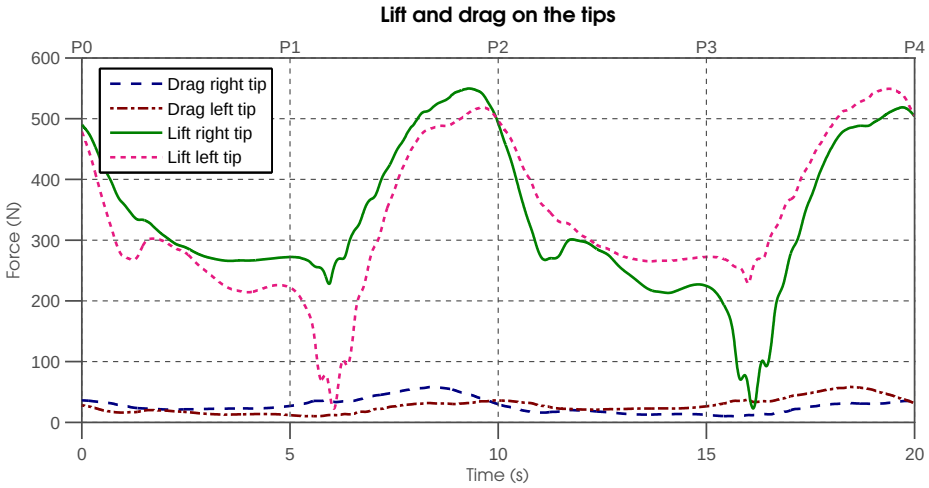


Figure 9-9: The lift and drag force on the tips of the kite while flying a figure eight.

9-2-4 Kite speeds

The kite gains a low speed in the negative y-direction when it is steered to the right in the first seconds as can be seen in Figure 9-10. This is according to what is described by Breukels [7] and a result from the higher lift force on the right tip than on the left tip.

The speed in the z-direction is due to the frame definition. The z-axis of the finite element model is not defined parallel to the tether line, but slightly tilted. This results in a velocity component in the z-direction which is part of the forward velocity. The actual velocity in the z-direction is almost zero, because the tether length is fixed.

Figure 9-10 shows that the forward speed of the kite varies roughly between 18m/s and 28 m/s ($v_{forward} = \sqrt{v_x^2 + v_z^2}$). van den Heuvel [48] shows that the maximum flying speed of a kite while flying crosswind is approximately the lift over drag ratio L/D times the wind velocity. A normal L/D for kites of this type is around 5. The wind speed is 5m/s, meaning a maximum kite speed of 25m/s. The obtained speeds are a bit higher than expected, but in the same order. The main factor that influences the speed is the lift over drag ratio L/D of the kite and is indeed too high as will be discussed in Section 9-3. The highest velocity is obtained between P_1 and P_2, which makes sense because there it reaches its lowest point and converted all the potential energy to kinetic energy.

The rotational speeds in Figure 9-11 show that the pitch angle stays almost constant. It will only change due to the action of the power controller to keep the power ratio in the kite constant. The rolling movements of the kite are also very small. When the kite makes a sharp corner, the kite also starts to roll a bit. The highest yawing rates are obtained right after passing point P_1. The curvature of the figure eight there is the sharpest and the velocity of the kite high. As a result the yawing speed needs to be high to follow the trajectory.

Besides being somewhat too high, the velocities show behaviour as expected.

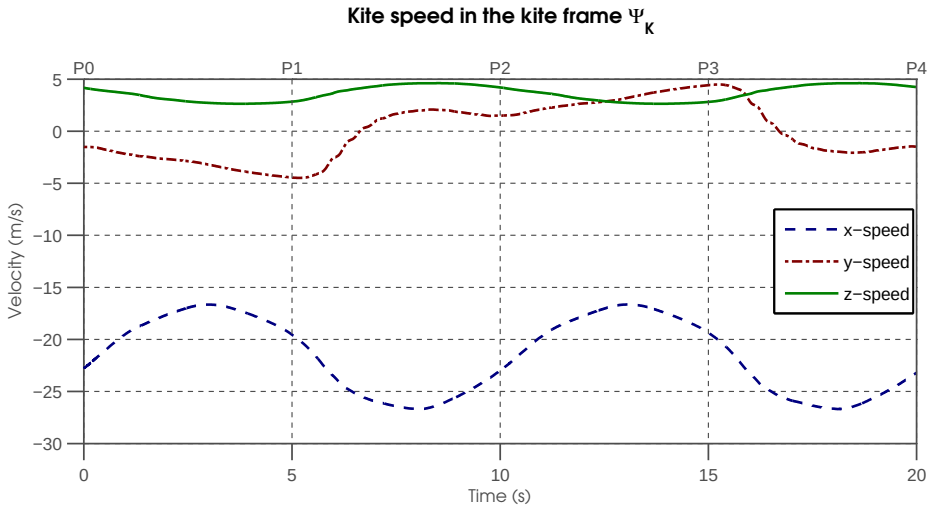


Figure 9-10: The speed of the kite while flying a figure eight.

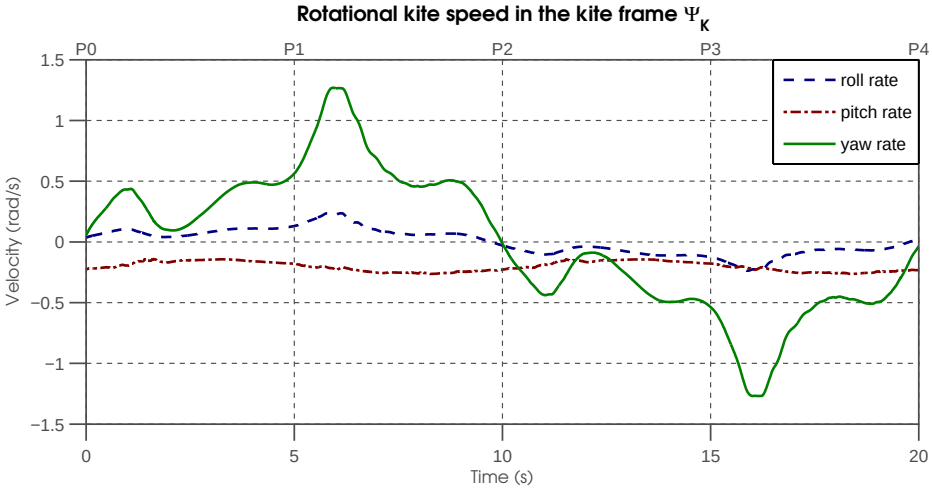


Figure 9-11: The rotational speed of the kite while flying a figure eight.

9-2-5 Line forces

As expected, the force in the right steering line is higher during the first 10 seconds of the flight while cornering to the right as shown in Figure 9-12 and vice versa between 10-20 seconds. The total pulling force of the kite on the tether reaches almost 6000N, which is a very high force, but also realistic. This is in the same order of magnitude compared to experimental data from test flights with the real system with comparable kites. The difference with the real system is that the tether is reeled out during the powering phase, significantly decreasing the tether force. The reel out velocity can be controlled to maintain a certain force in the tether. Since reeling out is not implemented in the model, a one to one comparison with available test data cannot be made. Forces are expected to be slightly too high, for the same reason that the velocities are too high.

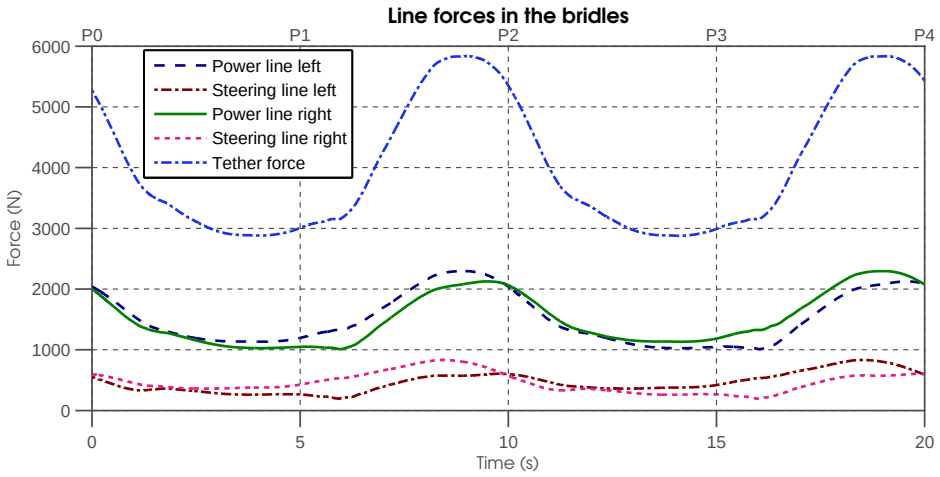


Figure 9-12: The bridle and tether forces while flying a figure eight.

9-3 Aerodynamics

The lift over drag ratio L/D was calculated for every time step in the trajectory by calculating the total aerodynamic forces on the kite and projecting them parallel and perpendicular to the apparent wind vector. Figure 9-13 shows that the L/D ratio varies roughly between 7.5 and 9, while a normal value would be around 5 for this type of kite. This explains the high speeds that were obtained. Two probable explanations can be given. Firstly the aerodynamic model is not very accurate, it neglects tip vortices and other effects that would increase the drag. A second reason comes from the modelling choices that were made in the finite element model. The canopy is not attached to the top of the leading edge, but connected to the middle of the leading edge beams. This results in a much lower camber of the airfoil. A lower camber lowers the drag force more significantly than the lift force, resulting in a higher L/D ratio. Furthermore the finite element model is somewhat stiffer than the real kite due to the used coarse mesh, also resulting in less deformation of the canopy and a smaller camber.

The aerodynamics are the main source of uncertainties in the model. All the results completely depend on the forces from the aerodynamic model. If those are not accurate, the end results will not be accurate. Much can be gained from improving the aerodynamic model.

Figure 9-14 shows that the power controller keeps the ratio of the forces between the steering lines and the power lines around 0.3. The controller has a low gain and was never expected to keep the ratio at exactly 0.3, so these results show the wanted behaviour.

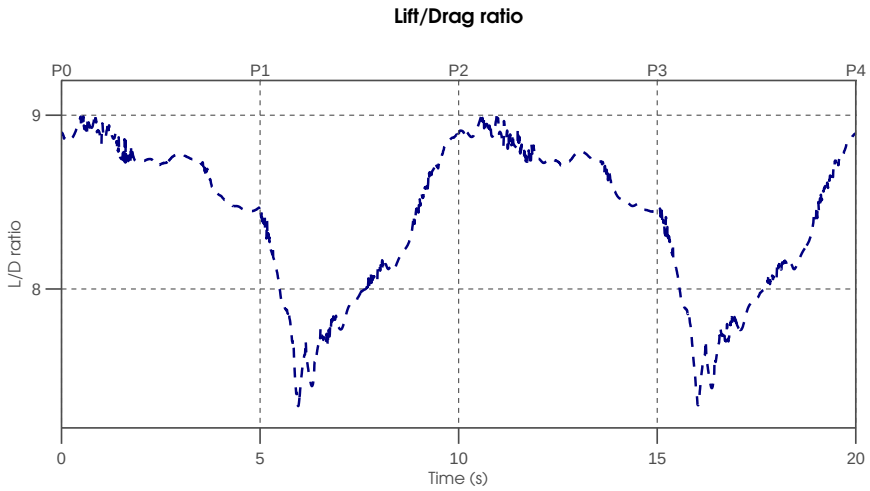


Figure 9-13: The lift/drag ratio L/Dwhile flying a figure eight.

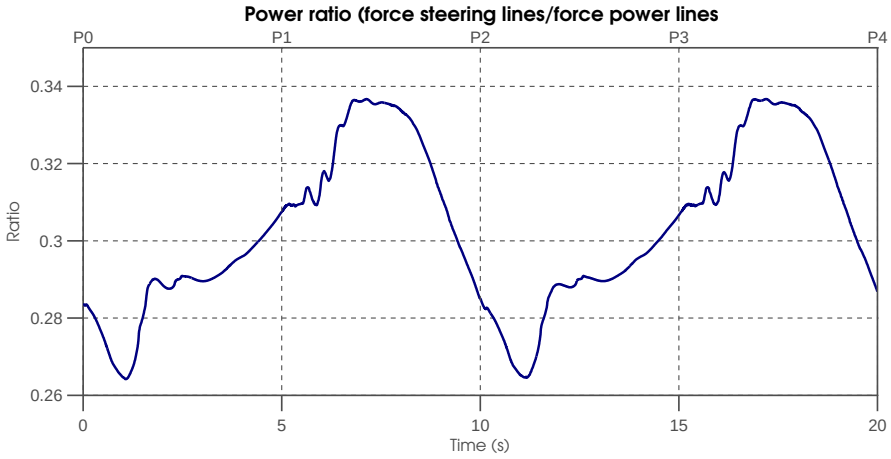


Figure 9-14: The power ratio (force steering lines/force power lines) while flying a figure eight.

9-4 Structural deformation

The kite shows more typical deformation modes besides the steering deformation. Figure 9-15 shows the displacements of the four bridle points in the y direction of the kite frame. It can be seen that the bridle points A and B move to and from point C and D. This typically happens in the real kite as well and is sometimes referred to as jellyfishing as shown in Figure 9-16. This is another realistic structural deformation of the kite.

The maximum deformation of the nodes at both the tips is approximately 0.7 meters. This is a total displacement of 1.4 meters with a span of the kite of 5.8 meters, resulting in a deformation of almost 25 percent. This clearly shows the need for the non linear modelling assumptions.

The displacements during the flight are smaller. Each bridle attachment points stays within a region of 0.3 meters in the y-direction and the kite acts quite stable during the simulations without much oscillations. Real kites also show this behaviour. Their nature is very flexible,

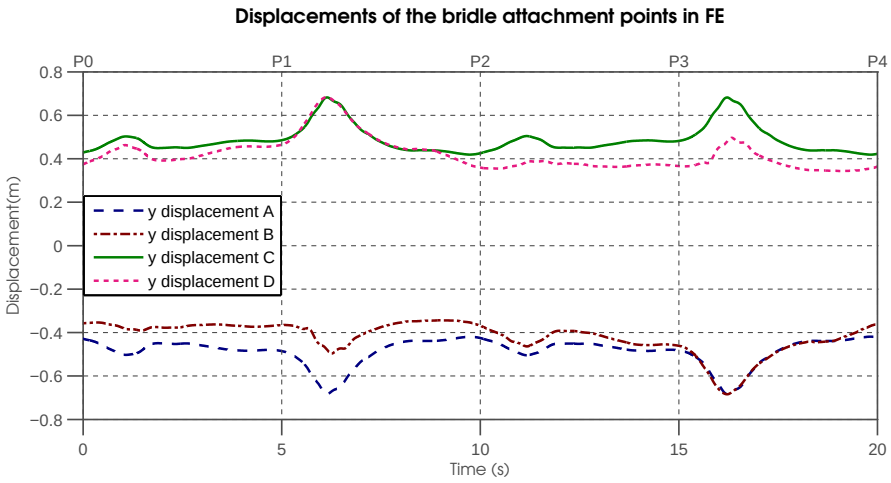


Figure 9-15: The displacements of the bridle attachment points in the y-direction while flying a figure eight.

but the aerodynamic forces make it a structure that almost acts as if it were rigid. During sudden wind gusts the flexible behaviour of the kites reveals itself. Since the simulated wind in the model was constant, such behaviour was not expected to show up, but it would be interesting to investigate those effects in the future. Also, the non linear material behaviour of the inflatable beams was not implemented, which would probably result in some different flexible modes. The course finite element mesh also introduces extra stiffness that contributes to the total stiffness of the system.

Appendix C shows more pictures of deformations of the kite. Similar deformation patterns can be seen in the model, confirming its validity. Movies of the real flying kite also show similar oscillations at the tips and other dynamic behaviour. This movie material is difficult to use as real validation confirmation, but contributes to the overall idea that the kite model represents the global dynamic behaviour. More in depth studies of the dynamic behaviour need to be performed to fully validate this dynamic behaviour.

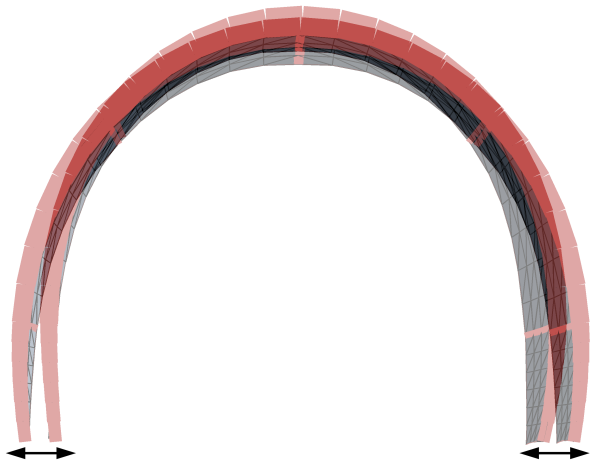


Figure 9-16: Typical jellyfishing motion that appears in the kite while flying.

9-5 Time integration

The time steps used by the dynamic Runge-Kutta explicit time integration algorithm are shown in Figure 9-17. Small time steps are needed in the transitional phase between turning to the right and turning to the left, where the yaw rate goes from positive to negative. Large time steps are taken around P_1 and P_2. Right after these points the highest steering input is applied to get a peak in the yawing rate. At these points the angles of attack at the tips show some oscillations and the lift force at one tips changes sign. This is were the time step size also starts to oscillate, it needs a small time step when the force direction is suddenly reversed. The average time step for this trajectory is 4.5 milliseconds and the speed of the simulation is between 25 and 30 times slower than real time, which is fast for interpreted MATLAB code and a simulation of this size.

Figure 9-17: The used time step by the dynamic ODE45 time integration algorithm while flying a figure eight.

9-6 Concluding remarks

The results presented in this section show a couple of things.

- The cornering behaviour of the kite model acts natural and is comparable to the model of Breukels [7] and the results obtained from experimental test data by Erhard and Strauch [13] and Jehle [23].
- Only a finite number of deformation modes has been observed, making this an interesting case for further reduction algorithms.
- The speed of the algorithm is already quite fast, but compiling the code would even further improve the speed.
- The combination of modelling choices and the aerodynamic model inaccuracies leads to an overestimation of the lift over drag ratio L/D resulting in too high velocities and forces.

Chapter 10

Conclusions & Recommendations

10-1 Conclusions

The goal of this research was to show a new realistic and reduced approach to model flying flexible inflatable tube kites used in airborne wind energy systems. The proposed method exist of three main building blocks: a quasi static finite element model of the kite, a dynamic simulation of the bridles and the tether and a quasi-static aerodynamic model. The kite model and aerodynamic model together form a fluid-structure-interaction (FSI) problem that needs to be solved iteratively and returns forces that are exerted on the bridle attachment points in the dynamic simulation of the bridles and tether.

Three reduction principles are introduced without losing any information about the global dynamic behaviour. It is assuming that the local inertia of the kite can be neglected and that the kite responds in a quasi static manner to the aerodynamic forces. A coarse mesh of the canopy is used together with regular beam elements to model the inflatable beams. Furthermore it is assumed that the resulting forces from solving the FSI-problem can be held constant during a time-step.

The (sub)goals were stated in Chapter 3. Firstly the model was examined to see whether the results are *realistic* and the following conclusions can be drawn.

- The model is fully derived from physical principles by using the finite element approach with the real material properties. The highly non-linear behaviour of the kite was nicely captured by using geometrically non-linear finite elements.
- The global dynamics of the kite are captured very realistically. All the important deformation modes that can be seen with a real kite, are seen in the model: an asymmetric twist while cornering and a jellyfish motion when subjected to a force disturbance.
- Results show that the kite model can steer by using the steering lines. The deformation of the kite model is comparable to a real kite. By pulling a steering line the kite deforms asymmetrically, creating an offset between the working lines of the lift forces on both tips resulting in a yaw moment and cornering of the kite.

- The deformation of the kite and the dynamics of the full model are dominated by the forces produced by the aerodynamic model. The used aerodynamic model of Breukels [7] was sufficiently accurate to prove this modelling approach, but has some shortcomings that have their direct effect on the validity of this model. Firstly the model could not be used to apply aerodynamic damping to the kite, because including the local speeds of the wing sections resulted in unstable simulations. Extra damping in the dynamic bridle model between the bridle attachment points was added to simulate this damping. Secondly the model neglects some effects as tip vortices and other three dimensional effects, resulting in an overestimation of the lift over drag ratio L/D, leading to higher speeds and forces than expected.

- The kite has multiple flying regimes. The dynamics of a cross wind flight are completely different from a stalling situation. The approach showed to work very well for crosswind flights, but had difficulties simulating stall or extreme loading situations. In those situations the canopy starts to experience more difficult dynamic behaviour such as fluttering (buckling) or beams start to buckle. More refined calculation algorithms are necessary to be able to solve those situations accurately. The model was primarily designed for the crosswind flying regime.

- The realistic behaviour of the kite makes it an interesting tool for kite designers to evaluate new designs before building them.

Secondly the model was examined on the *reduction* criteria.

- The kite was assumed to respond quasi-statically to the aerodynamic forces to create a fast model, thereby neglecting all the local dynamic effects and using the kite and aerodynamic model together (FSI-problem) as an external force generator to the dynamic tether model. This approach greatly reduced the calculation time. The dynamic numerical integration only has to be performed on 12 degrees of freedom instead of thousands (depending on the mesh size). This was a legitimate assumption, since the results show a realistic model and no important dynamic effects are missed.

- The same conclusion can be drawn about the assumption that the calculated forces from the FSI-problem can be kept constant during a time step and that the solving of the FSI-problem does not need to be included in the dynamic differential equations. Not having to solve the full FSI- problem multiple times per time step in the time integration algorithm speeds up the calculations by a factor 4 (depending on the algorithm).

- The main modelling choice to reduce the finite element model was to model the inflatable beams with regular non-linear beam elements, resulting in a massive reduction of degrees of freedom in the model. The properties of the inflatable beams can be transferred to normal beam elements without losing the correct dynamic behaviour.

- The reduction choices led to a model that is 25-30 slower than real time. This is very fast, considering the amount of calculations that need to be performed simultaneously in the a finite element model, FSI-problem and dynamic simulation all together. MATLAB has been used as programming environment, slowing things down quite drastically, because the code is not compiled. The speed of the model can probably be increased by a factor

5-10 by writing it in a compiled language. This makes it a good tool for controller design or optimization studies.

- The simulation shows a finite amount of deformation modes. This makes the model a good candidate for further new model reduction techniques as Proper Orthogonal Decomposition (POD) or the Discrete Empirical Interpolation Method (DEIM).

Some additional conclusions can be drawn considering the modelling approach.

- The developed method is general applicable and flexible. It is easy to model different kites with this method, even kites with a complex bridle system. That will only result in some extra degrees of freedom in the dynamic model. The mass distribution of a complex bridled kite will even be better than in the used C-shaped kite, where all the mass of the kite is only distributed over the four bridle attachment points.
- The approach is platform independent and does not rely on commercial software, although it could also be used in combination with existing finite element software, resulting in a very powerful approach.
- This thesis shows a practical approach to combine several building blocks. Building blocks can easily be changed, for example by introducing a new aerodynamic model. The approach is well balanced in a way that all components use a same level of detail.

All together it can be concluded that the new proposed approach is a successful one that fulfills the most important requirements and starts a new generation of more detailed kite models. Some details in the modelling are open for improvements to create an even more realistic model. It was outside the scope of this thesis, but an extensive model validation has to be performed to be able to fully verify the model. Hopefully this will lead to better kites that can be used to generate more energy from all the available wind high in the sky.

10-2 Recommendations

Some recommendations follow from the conclusions.

- The aerodynamic model is currently the limitation to the validity of the model. Better, fast aerodynamic models are needed to produce more realistic results.
- The tether reel in/out should be added to compare the model to experimental data.
- The kite model can be coupled to a flexible tether model to produce more realistic results when flying with long tethers.
- The Newton-Raphson convergence algorithm can be further optimized to converge faster and more stable for difficult situations as buckling of the canopy.
- The finite element model can be improved to capture the kite behaviour even better:
    1. The canopy needs to be attached to the top of the leading edge beams instead of to the middle. This creates a more realistic structural model and gives a better representation of the wing sections resulting in more realistic aerodynamic forces.
    2. The non linear material behaviour of the inflated beams should be implemented.

Appendix A

Geometric non-linear beam development

In this appendix, a geometric non-linear two node straight three dimensional beam element is developed that is used to model the inflatable leading edge and struts in the kite. The two dimensional approaches of Reddy [39] and Crisfield [10] are combined and extended to a three dimensional element in the Total Lagrangian framework. The non-linearity comes from the inclusion of the in-plane forces that are proportional to the square of the rotation of the transverse normal to the beam axis. The following (kinematic) assumptions are used:

- The element is based on the classic Euler-Bernoulli theory.
- The plane sections perpendicular to the axis of the beam remain plane, rigid and rotate such that they remain perpendicular to the deformed axis after deformation.
- Therefore the Poisson effect and transverse stresses can be neglected.
- Large transverse displacements, small strains and small to moderate rotations are assumed.

The coordinate system of the three-dimensional beam is depicted in Figure A-1. The coordinates are described by x, y and z and the displacement field of the beam by $u(x, y, z)$, $v(x, y, z)$ and $w(x, y, z)$. The element degrees of freedom are described by

$$\boldsymbol{q}_{el}^T = \begin{bmatrix} u_1 & v_1 & w_1 & \psi_{x1} & \psi_{y1} & \psi_{z1} & u_2 & v_2 & w_2 & \psi_{x2} & \psi_{y2} & \psi_{z2} \end{bmatrix} \tag{A-1}$$

The angles and curvatures are related to the displacement fields by

$$\psi_x = \phi, \psi_y = -\frac{\mathrm{d}w}{\mathrm{d}x}, \psi_z = \frac{\mathrm{d}v}{\mathrm{d}x} \tag{A-2}$$

$$\kappa_x = \frac{\mathrm{d}\phi}{\mathrm{d}x}, \kappa_y = -\frac{\mathrm{d}^2w}{\mathrm{d}x^2}, \kappa_z = \frac{\mathrm{d}^2v}{\mathrm{d}x^2} \tag{A-3}$$

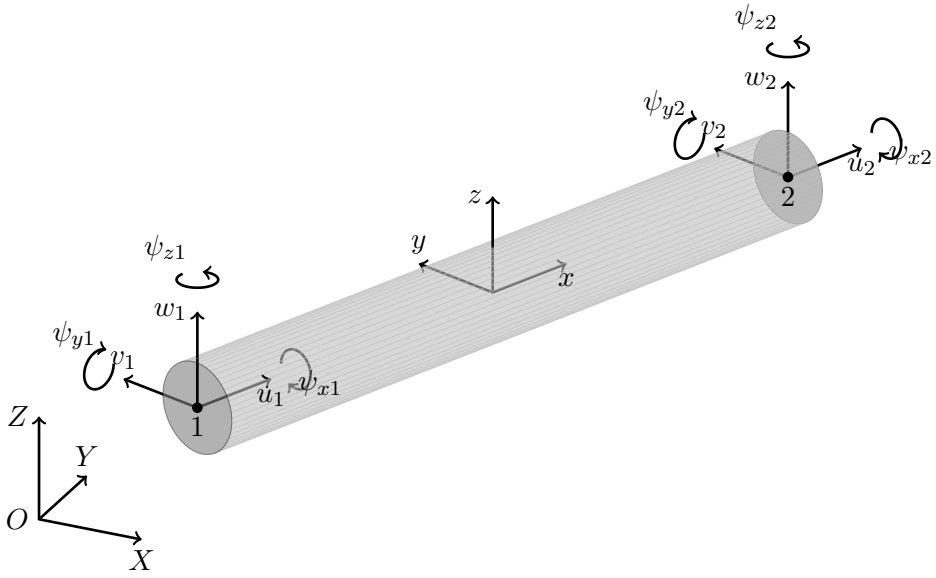


Figure A-1: The three dimensional two node geometrically non-linear beam element with elemental coordinate frame x, y and z and degrees of freedom $\boldsymbol{q}_{el}$.

From continuum mechanics the non-linear Green-Lagrange strains can be written as [39]

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{1}{2}\left(\frac{\partial u_m}{\partial x_i}\frac{\partial u_m}{\partial x_j}\right) \tag{A-4}$$

Omitting large strains, but keeping the square of the rotations of a transverse normal line in the beam, leads to the following strain in the axial direction, also known as the *von Kármán strains*. All the other strains are zero, because of the assumptions that were made. The non-linear contribution comes from this equation.

$$\varepsilon_{11} = \varepsilon_x = \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right) + \frac{1}{2}\left(\frac{\mathrm{d}v}{\mathrm{d}x}\right)^2 + \frac{1}{2}\left(\frac{\mathrm{d}w}{\mathrm{d}x}\right)^2 \tag{A-5}$$

As shown in Figure A-2 the displacement field $u(x)$ can be expressed in the axial displacement $\bar{u}$ and y and z coordinate as

$$u = \bar{u} - z\frac{\mathrm{d}w}{\mathrm{d}x} + y\frac{\mathrm{d}v}{\mathrm{d}x} \tag{A-6}$$

This leads after substitution of Eq. A-6 in Eq. A-5 to

$$\begin{aligned}\varepsilon_x &= \left(\frac{\mathrm{d}\bar{u}}{\mathrm{d}x}\right) + \frac{1}{2}\left(\frac{\mathrm{d}v}{\mathrm{d}x}\right)^2 + \frac{1}{2}\left(\frac{\mathrm{d}w}{\mathrm{d}x}\right)^2 - z\frac{\mathrm{d}^2w}{\mathrm{d}x^2} + y\frac{\mathrm{d}^2v}{\mathrm{d}x^2} \\ &= \bar{\varepsilon} + z\kappa_y + y\kappa_z\end{aligned} \tag{A-7}$$

For the torsion a linear approach is taken as described in Reference [19]. The shear strain γ can be written as

$$\gamma = \rho\frac{\mathrm{d}\phi}{\mathrm{d}x} = \rho\kappa_x \tag{A-8}$$

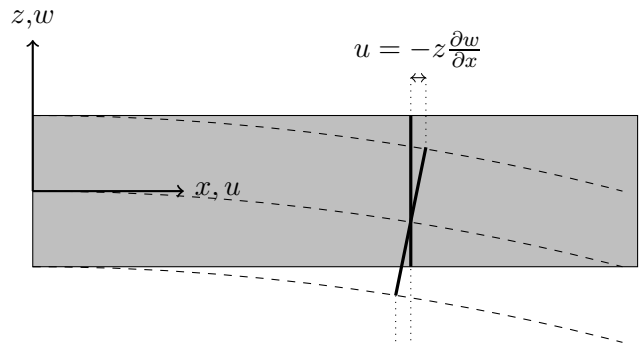


Figure A-2: The displacements due to bending in the zx-plane expressed by the rotation around the y-axis $\frac{\mathrm{d}w}{\mathrm{d}x}$.

with ρ the radial distance from the axis of the beam.

The weak form of structural problems can now be derived by using the virtual work principle. The virtual strain energy can be written as

$$\delta\mathcal{V}_{int} = \int_V (\delta\varepsilon_{ij}\sigma_{ij} + \delta\gamma\tau)\mathrm{d}V \tag{A-9}$$

$$= \int_x \int_A (\delta\varepsilon_{11}\sigma_{11} + \delta\gamma\tau)\mathrm{d}A\mathrm{d}x \tag{A-10}$$

with the strains ε_{ij}, the stresses σ_{ij}, the shear strain γ and shear stress τ.
Since in general holds that

$$\delta\varepsilon = \frac{\partial\varepsilon}{\partial u}\delta u + \frac{\partial\varepsilon}{\partial v}\delta v + \frac{\partial\varepsilon}{\partial w}\delta w \tag{A-11}$$

the variation of the strains as described in Eq. A-7 and Eq. A-8 results in

$$\delta\varepsilon_{11} = \frac{\mathrm{d}\delta\bar{u}}{\mathrm{d}x} + \frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}\delta v}{\mathrm{d}x} + \frac{\mathrm{d}w}{\mathrm{d}x}\frac{\mathrm{d}\delta w}{\mathrm{d}x} + z\delta\kappa_y + y\delta\kappa_z \tag{A-12}$$

$$\delta\gamma = \rho\delta\kappa_x \tag{A-13}$$

Substituting this in Eq. A-9 results in

$$\delta\mathcal{V}_{int} = \int_x \left(N\left(\frac{\mathrm{d}\delta\bar{u}}{\mathrm{d}x} + \frac{\mathrm{d}v}{\mathrm{d}x}\frac{\mathrm{d}\delta v}{\mathrm{d}x} + \frac{\mathrm{d}w}{\mathrm{d}x}\frac{\mathrm{d}\delta w}{\mathrm{d}x}\right) + M_y\delta\kappa_y + M_z\delta\kappa_z + M_x\delta\kappa_x \right)\mathrm{d}x \tag{A-14}$$

with

$$N = \int_A \sigma_{11}\mathrm{d}A = EA\bar{\varepsilon} \qquad M_y = \int_A \sigma_{11}z\mathrm{d}A = EI_y\kappa_y \tag{A-15}$$

$$M_z = \int_A \sigma_{11}y\mathrm{d}A = EI_x\kappa_z \qquad M_x = \int_A \tau\rho dA = GJ_x\kappa_x \tag{A-16}$$

where the constitutive relations assuming elastic properties were used with the Young's modulus E, shear modulus G and poisson ratio v.

$$\sigma = E\varepsilon \tag{A-17}$$

$$\tau = G\gamma \tag{A-18}$$

$$G = \frac{E}{2(1+v)} \tag{A-19}$$

Linear shape function are used to describe the displacement field $\bar{u}$ and torsional displacements ϕ and cubic polynomial shape functions are used to describe the displacement fields w and v. The displacement fields are described by the shape functions $\boldsymbol{N}_i$ and the values at node 1 and node 2 represented by $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ and $\boldsymbol{\phi}$

$$\boldsymbol{u}^T = [u_1 \ u_2] \tag{A-20}$$
$$\boldsymbol{v}^T = [v_1 \ \psi_{z1} \ v_2 \ \psi_{z2}] \tag{A-21}$$
$$\boldsymbol{w}^T = [w_1 \ \psi_{y1} \ w_2 \ \psi_{y2}] \tag{A-22}$$
$$\boldsymbol{\phi} = [\psi_{x1} \ \psi_{x2}] \tag{A-23}$$

The displacement functions become

$$\bar{u} = \boldsymbol{N}_u(\xi)\boldsymbol{u} \tag{A-24}$$
$$v = \boldsymbol{N}_v(\xi)\boldsymbol{v} \tag{A-25}$$
$$w = \boldsymbol{N}_w(\xi)\boldsymbol{w} \tag{A-26}$$
$$\phi = \boldsymbol{N}_\phi(\xi)\boldsymbol{\phi} \tag{A-27}$$

Let us introduce the non-dimensional variable ξ to describe the coordinate along the element

$$\xi = \frac{2}{L}x - 1 \tag{A-28}$$

The shape functions become

$$\boldsymbol{N}_u^T = \boldsymbol{N}_\phi^T = \frac{1}{2}\begin{bmatrix} 1-\xi \\ 1+\xi \end{bmatrix} \qquad \boldsymbol{N}_v^T = \begin{bmatrix} \frac{1}{4}(1-\xi)^2(2+\xi) \\ \frac{1}{8}L(1-\xi)^2(1+\xi) \\ \frac{1}{4}(1+\xi)^2(2-\xi) \\ -\frac{1}{8}L(1+\xi)^2(1-\xi) \end{bmatrix} \qquad \boldsymbol{N}_w^T = \begin{bmatrix} \frac{1}{4}(1-\xi)^2(2+\xi) \\ -\frac{1}{8}L(1-\xi)^2(1+\xi) \\ \frac{1}{4}(1+\xi)^2(2-\xi) \\ \frac{1}{8}L(1+\xi)^2(1-\xi) \end{bmatrix} \tag{A-29}$$

Differentiation of the shape functions leads to

$$\frac{\mathrm{d}\bar{u}}{\mathrm{d}x} = \begin{bmatrix} -1/L \\ 1/L \end{bmatrix}^T \boldsymbol{u} = \boldsymbol{b}_u\boldsymbol{u} \tag{A-30}$$

$$\boldsymbol{b}_\phi = \boldsymbol{b}_u \Rightarrow \phi' = \boldsymbol{b}_\phi\boldsymbol{\phi} \tag{A-31}$$

$$\frac{\mathrm{d}v}{\mathrm{d}x} = \frac{1}{4L}\begin{bmatrix} 6(\xi^2-1) \\ L(3\xi^2-2\xi-1) \\ 6(1-\xi^2) \\ L(3\xi^2+2\xi-1) \end{bmatrix}^T \boldsymbol{v} = \boldsymbol{b}_v\boldsymbol{v} \tag{A-32}$$

$$\frac{\mathrm{d}^2v}{\mathrm{d}x^2} = \frac{1}{L^2}\begin{bmatrix} 6\xi \\ L(3\xi-1) \\ -6\xi \\ L(3\xi+1) \end{bmatrix} \boldsymbol{v} = \boldsymbol{c}_v\boldsymbol{v} \tag{A-33}$$

$$\frac{\mathrm{d}w}{\mathrm{d}x} = \frac{1}{4L}\begin{bmatrix} 6(\xi^2-1) \\ -L(3\xi^2-2\xi-1) \\ 6(1-\xi^2) \\ -L(3\xi^2+2\xi-1) \end{bmatrix}^T \boldsymbol{v} = \boldsymbol{b}_w \boldsymbol{v} \tag{A-34}$$

$$\frac{\mathrm{d}^2w}{\mathrm{d}x^2} = \frac{1}{L^2}\begin{bmatrix} 6\xi \\ -L(3\xi-1) \\ -6\xi \\ -L(3\xi+1) \end{bmatrix} \boldsymbol{w} = \boldsymbol{c}_w \boldsymbol{w} \tag{A-35}$$

$$\kappa_x = \boldsymbol{b}_\phi \boldsymbol{\phi} \qquad \kappa_y = -\boldsymbol{c}_w \boldsymbol{w} \qquad \kappa_z = \boldsymbol{c}_v \boldsymbol{v} \tag{A-36}$$

This leads after substitution in Eq. A-7 and A-8 to

$$\bar{\varepsilon} = \boldsymbol{b}_u \boldsymbol{u} + \frac{1}{2}(\boldsymbol{b}_v \boldsymbol{v})^2 + \frac{1}{2}(\boldsymbol{b}_w \boldsymbol{w})^2 \tag{A-37}$$

$$\gamma = \rho \boldsymbol{b}_\phi \boldsymbol{\phi} \tag{A-38}$$

The internal energy from Eq. A-14 and variation of the strains in Eq. A-12 can now be written as

$$\delta\varepsilon_{11} = \boldsymbol{b}_u \delta\boldsymbol{u} + (\boldsymbol{b}_v \boldsymbol{v})\boldsymbol{b}_v \delta\boldsymbol{v} + (\boldsymbol{b}_w \boldsymbol{w})\boldsymbol{b}_w \delta\boldsymbol{w} - z\boldsymbol{c}_w \delta\boldsymbol{w} + y\boldsymbol{c}_v \delta\boldsymbol{v} \tag{A-39}$$

$$\delta\gamma = \rho \boldsymbol{b}_\phi \delta\boldsymbol{\phi} \tag{A-40}$$

$$\begin{aligned} \delta\mathcal{V}_{int} &= \int_x \left(N\left(\boldsymbol{b}_u \delta\boldsymbol{u} + (\boldsymbol{b}_v \boldsymbol{v})\boldsymbol{b}_v \delta\boldsymbol{v} + (\boldsymbol{b}_w \boldsymbol{w})\boldsymbol{b}_w \delta\boldsymbol{w}\right) - M_y \boldsymbol{c}_w \delta\boldsymbol{w} + M_z \boldsymbol{c}_v \delta\boldsymbol{v} + M_x \boldsymbol{b}_\phi \delta\boldsymbol{\phi}\right) \mathrm{d}x \\ &= \boldsymbol{U}_i^T \delta\boldsymbol{u} + \boldsymbol{V}_i^T \delta\boldsymbol{v} + \boldsymbol{W}_i^T \delta\boldsymbol{w} + \boldsymbol{\Phi}_i^T \delta\boldsymbol{\phi} \end{aligned} \tag{A-41}$$

Let us write the internal force of an element as

$$\boldsymbol{f}_{int}^T = (\boldsymbol{U}^T, \boldsymbol{V}^T, \boldsymbol{W}^T, \boldsymbol{\Phi}^T) \tag{A-42}$$

with $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$, $\boldsymbol{\Phi}$ parts of the total internal energy from Eq. A-41 associated to the respective degrees of freedom

$$\boldsymbol{U} = \int N \boldsymbol{b}_u^T \mathrm{d}x \tag{A-43}$$

$$\boldsymbol{V} = \int (N(\boldsymbol{b}_v \boldsymbol{v})\boldsymbol{b}_v^T + M_z \boldsymbol{c}_v^T) \mathrm{d}x \tag{A-44}$$

$$\boldsymbol{W} = \int (N(\boldsymbol{b}_w \boldsymbol{w})\boldsymbol{b}_w^T - M_y \boldsymbol{c}_w^T) \mathrm{d}x \tag{A-45}$$

$$\boldsymbol{\Phi} = \int M_x \boldsymbol{b}_\phi^T \mathrm{d}x \tag{A-46}$$

The stiffness matrix now takes the form

$$\boldsymbol{K}_t = \begin{bmatrix} \boldsymbol{K}_{uu} & \boldsymbol{K}_{uv} & \boldsymbol{K}_{uw} & \boldsymbol{K}_{u\phi} \\ \boldsymbol{K}_{vu} & \boldsymbol{K}_{vv} & \boldsymbol{K}_{vw} & \boldsymbol{K}_{v\phi} \\ \boldsymbol{K}_{wu} & \boldsymbol{K}_{wv} & \boldsymbol{K}_{ww} & \boldsymbol{K}_{w\phi} \\ \boldsymbol{K}_{\phi u} & \boldsymbol{K}_{\phi v} & \boldsymbol{K}_{\phi w} & \boldsymbol{K}_{\phi\phi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{uu} & \boldsymbol{K}_{uv} & \boldsymbol{K}_{uw} & \varnothing \\ \boldsymbol{K}_{vu} & \boldsymbol{K}_{vv} & \boldsymbol{K}_{vw} & \varnothing \\ \boldsymbol{K}_{wu} & \boldsymbol{K}_{wv} & \boldsymbol{K}_{ww} & \varnothing \\ \varnothing & \varnothing & \varnothing & \boldsymbol{K}_{\phi\phi} \end{bmatrix} \tag{A-47}$$

and is obtained by differentiation of the internal force vector. Keep in mind that $N(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$,$M_z(\boldsymbol{v})$ and $M_y(\boldsymbol{w})$ are functions of different variables.

$$\boldsymbol{K}_{uu} = \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{u}} = \int \boldsymbol{b}_u^T \frac{\partial N}{\partial \boldsymbol{u}} \mathrm{d}x = \int EA \boldsymbol{b}_u^T \boldsymbol{b}_u \mathrm{d}x \tag{A-48}$$

$$\boldsymbol{K}_{uv} = \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{v}} = \int \boldsymbol{b}_u^T \frac{\partial N}{\partial \boldsymbol{v}} \mathrm{d}x = \int EA(\boldsymbol{b}_v \boldsymbol{v}) \boldsymbol{b}_u^T \boldsymbol{b}_v \mathrm{d}x \tag{A-49}$$

$$\boldsymbol{K}_{vu} = \boldsymbol{K}_{uv}^T \tag{A-50}$$

$$\boldsymbol{K}_{uw} = \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{w}} = \int \boldsymbol{b}_u^T \frac{\partial N}{\partial \boldsymbol{w}} \mathrm{d}x = \int EA(\boldsymbol{b}_w \boldsymbol{w}) \boldsymbol{b}_u^T \boldsymbol{b}_w \mathrm{d}x \tag{A-51}$$

$$\boldsymbol{K}_{wu} = \boldsymbol{K}_{uw}^T \tag{A-52}$$

$$\begin{aligned} \boldsymbol{K}_{vv} = \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{v}} &= \int \boldsymbol{c}_v^T \frac{\partial M_z}{\partial \boldsymbol{v}} + N \boldsymbol{b}_v^T \boldsymbol{b}_v + (\boldsymbol{b}_v \boldsymbol{v}) \boldsymbol{b}_v^T \frac{\partial N}{\partial \boldsymbol{v}} \mathrm{d}x \\ &= \int EI \boldsymbol{c}_v^T \boldsymbol{c}_v + N \boldsymbol{b}_v^T \boldsymbol{b}_v + EA(\boldsymbol{b}_v \boldsymbol{v})^2 \boldsymbol{b}_v^T \boldsymbol{b}_v \mathrm{d}x \end{aligned} \tag{A-53}$$

$$\begin{aligned} \boldsymbol{K}_{ww} = \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{w}} &= \int -\boldsymbol{c}_w^T \frac{\partial M_y}{\partial \boldsymbol{w}} + N \boldsymbol{b}_w^T \boldsymbol{b}_w + (\boldsymbol{b}_w \boldsymbol{w}) \boldsymbol{b}_w^T \frac{\partial N}{\partial \boldsymbol{w}} \mathrm{d}x \\ &= \int EI \boldsymbol{c}_w^T \boldsymbol{c}_w + N \boldsymbol{b}_w^T \boldsymbol{b}_w + EA(\boldsymbol{b}_w \boldsymbol{w})^2 \boldsymbol{b}_w^T \boldsymbol{b}_w \mathrm{d}x \end{aligned} \tag{A-54}$$

$$\begin{aligned} \boldsymbol{K}_{vw} = \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{w}} &= \int (\boldsymbol{b}_v \boldsymbol{v}) \boldsymbol{b}_v^T \frac{\partial N}{\partial \boldsymbol{w}} \mathrm{d}x \\ &= \int EA(\boldsymbol{b}_v \boldsymbol{v})(\boldsymbol{b}_w \boldsymbol{w}) \boldsymbol{b}_v^T \boldsymbol{b}_w \mathrm{d}x \end{aligned} \tag{A-55}$$

$$\boldsymbol{K}_{wv} = \boldsymbol{K}_{vw}^T \tag{A-56}$$

$$\boldsymbol{K}_{\phi\phi} = \frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{\phi}} = \int GJ_x \boldsymbol{b}_\phi^T \boldsymbol{b}_\phi \mathrm{d}x \tag{A-57}$$

The symbolic toolbox in MATLAB was used to integrate the equations for the internal forces and stiffness matrices.

Appendix B

Finite element implementation verification

B-1 Introduction

In this appendix the in MATLAB developed shell element used to model the canopy and the beam element used to model the leading edge and the struts will be verified.

Obtained solutions from the MATLAB elements are compared to solutions obtained from the finite element package Ansys. This option was preferred over comparison with some literature cases, since this gives more flexibility in the choice of the load cases. Furthermore Ansys is widely used finite element software and can therefore be trusted to generate reliable solutions for these basic elements.

To verify the correctness of the linear and non-linear implementation of the stiffness matrix, both element types will be subjected to different load cases. The maximum deflection and mean error scaled by the maximum deflection $\frac{|\boldsymbol{q}_{Matlab}-\boldsymbol{q}_{Ansys}|}{N_{dof}*q_{max}}$ will be used as a measure to compare the results from Ansys and MATLAB .

B-2 Shell elements

B-2-1 Set-up

For the tests with shell elements an aluminium plate with dimensions 1m x 0.5m x 0.001m will be used with a Young's modulus of 70GPa. The plate has a structured mesh with 400 elements, 231 nodes and therefore 1386 degrees of freedom.

In MATLAB the plate will be modelled with the shell elements as described in Chapter 5. In Ansys there are different shell element types, but only the SHELL63 element is capable of using a triangular element with three nodes. This shell has both bending and membrane

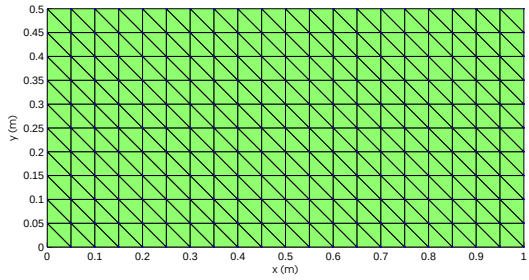


Figure B-1: Plate set-up for the shell element verification.

capabilities, six degrees of freedom for each node and geometric non-linear options. The theory behind this element is different from the theory behind the MATLAB model, so some differences in the results are expected. The SHELL63 element is for example capable of handling large deformations, where the MATLAB element was designed for moderate deformations.

B-2-2 Load cases

Load case 1

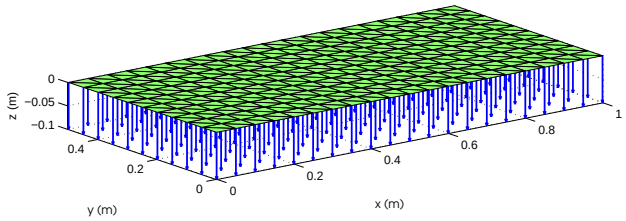


Figure B-2: Load case 1: setup

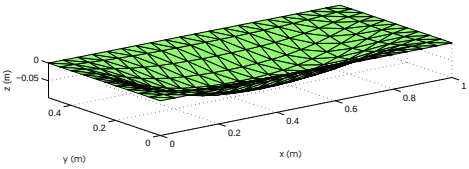
Load case: as in Figure B-2.
Boundary conditions: clamped on all sides.
Load: $F_z = -10N$ for each node.
Results: Table B-1 and Figure B-3.

Table B-1: Results load case 1 shell element

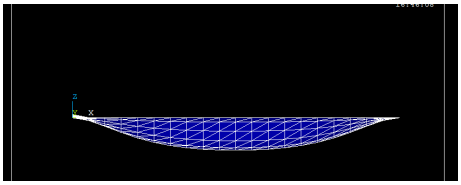
	MATLAB	Ansys	Error
Linear	0.0974m	0.0995m	0.0096
Non-linear	0.00522m	0.0519m	0.0375

Load case 2

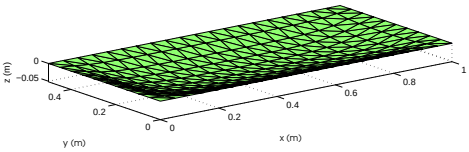
Load case: same as for load case 1, except the direction of the forces.
Boundary conditions: clamped on all sides.
Load: $F_x = 10N$ for each node.
Results: Table B-2.



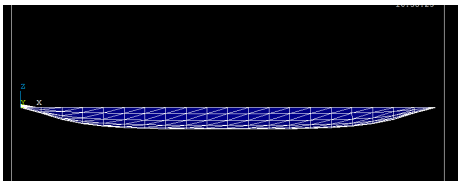
(a) Linear, MATLAB , deformation scale = 1, maximum deformation = 0.0974m



(b) Linear, Ansys, deformation scale = 1, maximum deformation = 0.0995m



(c) Non-linear, MATLAB , deformation scale = 10, maximum deformation = 0.00522m



(d) Non-linear, Ansys, deformation scale = 10, maximum deformation = 0.00519m

Figure B-3: The results for load case 1 for the shell element.

Table B-2: Results load case 2 shell element

	MATLAB	Ansys	Error
Linear	3.2401×10^{-6}m	3.2654e-$\times10^{-6}$m	0.111
Non-linear	3.2401×10^{-6}m	3.2654e-$\times10^{-6}$m	0.111

Load case 3

Load case: same as for load case 1, except the direction of the forces
Boundary conditions: clamped on all sides.
Load: $F_y = 10N$ for each node.
Results: Table B-3.

Table B-3: Results load case 3 shell element

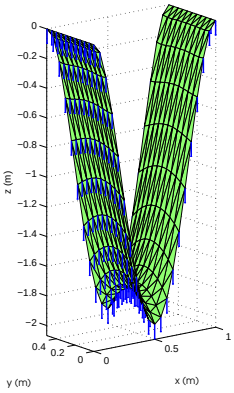
	MATLAB	Ansys	Error
Linear	1.6401×10^{-6}m	1.6393×10^{-6}m	0.0665
Non-linear	1.6401×10^{-6}m	1.6393×10^{-6}m	0.0665

Load case 4

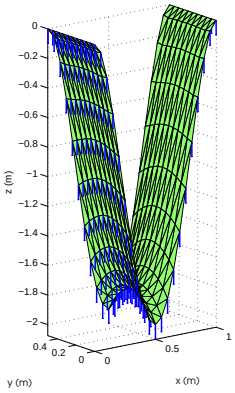
Load case: same as for load case 1, except the boundary conditions.
Boundary conditions: clamped on two smaller sides (sides in y-direction).
Load: $F_z = -10N$ for each node.
Results: Table B-4 and Fig B-4.

Table B-4: Results load case 4 shell element

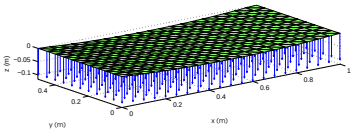
	MATLAB	Ansys	Error
Linear	1.9678m	1.9739m	0.0017
Non-linear	0.0161m	0.0160m	0.0168



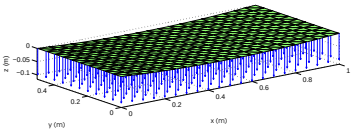
(a) Linear, MATLAB , deformation scale = 1, maximum deformation = 1.9678m



(b) Linear, Ansys, deformation scale = 1, maximum deformation = 1.9739m



(c) Non-linear, MATLAB , deformation scale = 1, maximum deformation = 0.0161m



(d) Non-linear, Ansys, deformation scale = 1, maximum deformation = 0.0160m

Figure B-4: The results for load case 4 for the shell element.

Load case 5

Load case: curved shell as in Fig B-5.
Boundary conditions: clamped on two smaller sides (sides in y-direction).
Load: $F_z = 500N$ for node 135.
Results: Table B-5 and Fig B-6.

Table B-5: Results load case 5 shell element

	MATLAB	Ansys	Error
Linear	0.1456m	0.1472m	0.0066
Non-linear	0.0478m	0.0558m	0.1303

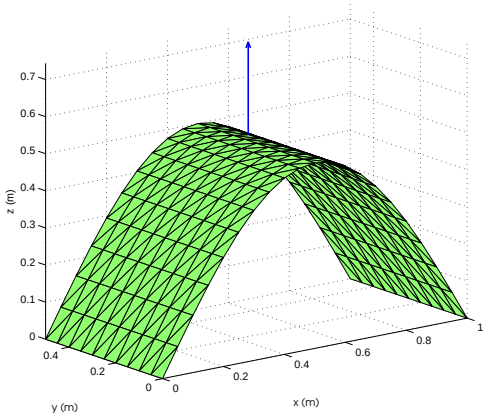


Figure B-5: LoadCase 5: setup

Load case 6

Load case: curved shell as in Fig B-5.
Boundary conditions: clamped at the four corners.
Load: $F_z = 10N$ for all nodes.
Results: Table B-6 and Fig B-7.

Table B-6: Results load case 6 shell element

	MATLAB	Ansys	Error
Linear	0.1344m	0.1332m	0.0126
Non-linear	0.0124m	0.0126m	0.0231

Load case 7

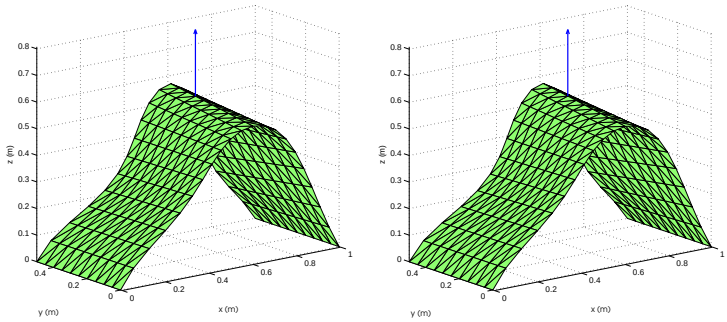
Load case: curved shell as in Fig B-5, but the material properties changed to $t = 0.8 \times 10^{-3} m$, $E = 0.68GPa$, $\rho = 625kg/m^3$.
Boundary conditions: hinged on the two smaller sides (y-direction).
Load: $F_z = 0.2N$ on all nodes.
Results: Table B-7 and Fig B-8.

Table B-7: Results load case 7 shell element

	MATLAB	Ansys	Error
Linear	0.4224m	0.4234m	0.0073
Non-linear	0.01384m	0.01378m	0.0243

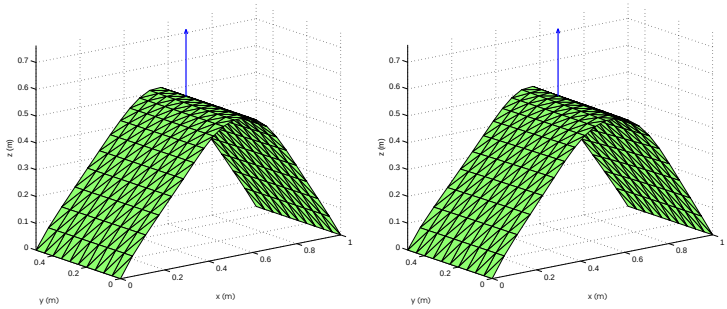
Load case 8

Load case: curved shell with less elements and a different load as in Fig B-9.
Boundary conditions: hinged on the two smaller sides (y-direction).



(a) Linear, MATLAB , deformation scale = 1, maximum deformation = 0.1456m

(b) Linear, Ansys, deformation scale = 1, maximum deformation = 0.1472m



(c) Non-linear, MATLAB , deformation scale = 1, maximum deformation = 0.0478m

(d) Non-linear, Ansys, deformation scale = 1, maximum deformation = 0.0558m

Figure B-6: The results for load case 5 for the shell element.

Load: $F_z = 6N$ on node 61.
Results: Table B-8 and Fig B-10.

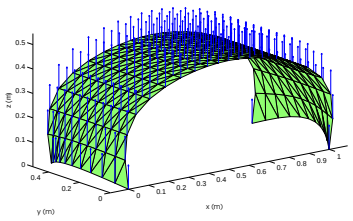
Table B-8: Results load case 8 shell element

	MATLAB	Ansys	Error
Linear	0.7203m	0.7310m	0.0102
Non-linear	0.0808m	0.0875m	0.1017

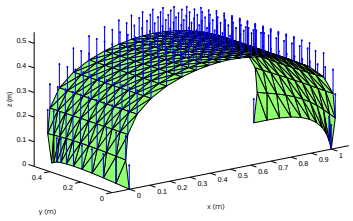
B-2-3 Conclusions

The following can be concluded from the load tests with the shell elements.

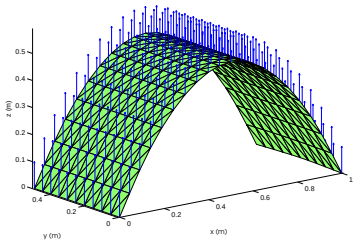
- In all the out of plane load cases the difference between the linear and non-linear case is quite significant. This is due to the fact that the undeformed thin plate has almost no bending stiffness. When the plate bends, more of the forces are taken by the in plane stiffness. The geometrically non-linear element recalculates the stiffness matrix to include this effect, the linear element doesn't. Load case 1 shows this effect clearly.



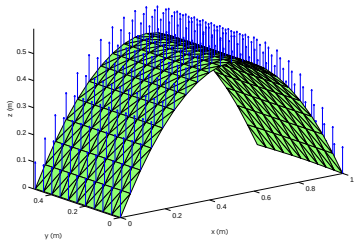
(a) Linear, MATLAB , deformation scale = 1, maximum deformation = 0.1344m



(b) Linear, Ansys, deformation scale = 1, maximum deformation = 0.1332m



(c) Non-linear, MATLAB , deformation scale = 1, maximum deformation = 0.0124m



(d) Non-linear, Ansys, deformation scale = 1, maximum deformation = 0.0126m

Figure B-7: The results for load case 6 for the shell element.

- In load case 2 and 3 it can be seen that the solution is the same for the linear and non-linear solution. This is because the deflections are so small that the tangential stiffness matrix doesn't need to be recalculated in the non-linear element. Both elements use the same stiffness matrix.

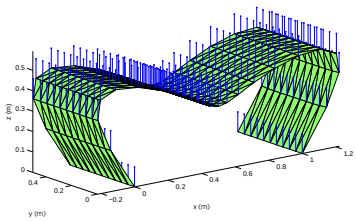
- In general the deflections and deformation shapes calculated by MATLAB and Ansys are quite similar.

- In some cases the deflections are larger than allowed for the MATLAB model, since it is optimized for moderate deformations.

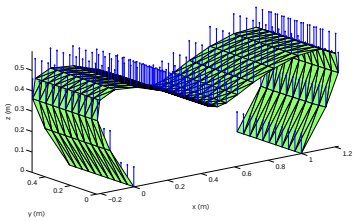
- Load case 5 shows quite a large difference for the non-linear solution. A probable cause is the applied point load. Such a load is expected to give inaccurate results, since you should never apply a point load to a plate or membrane, because this would in theory result in an infinite displacement, because it acts on an infinite small area.

- Load case 7 and 8 show that the results are also quite similar for a thinner plate, even though a point load is applied in case 8. This is important since the shells used in the kite are very thin as well.

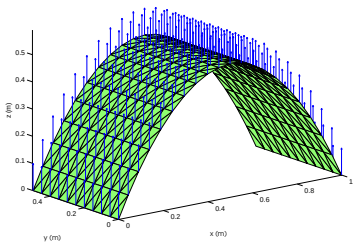
- It cannot be said whether the developed MATLAB element is stiffer or less tiff than the MATLAB element. This is different for each load case.



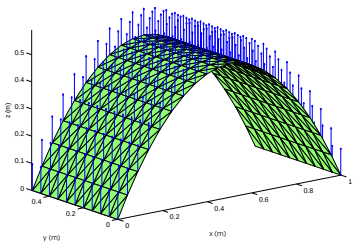
(a) Linear, MATLAB , deformation scale = 1, maximum deformation = 0.4224m



(b) Linear, Ansys, deformation scale = 1, maximum deformation = 0.4234m



(c) Non-linear, MATLAB , deformation scale = 1, maximum deformation = 0.01384m



(d) Non-linear, Ansys, deformation scale = 1, maximum deformation = 0.01378m

Figure B-8: The results for load case 7 for the shell element.

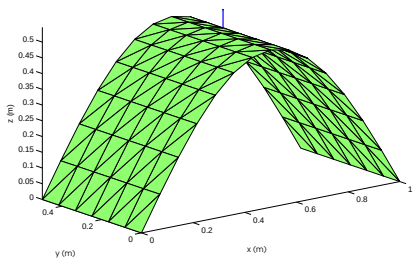
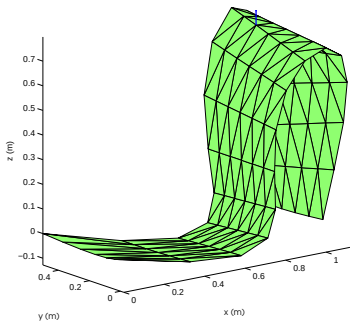
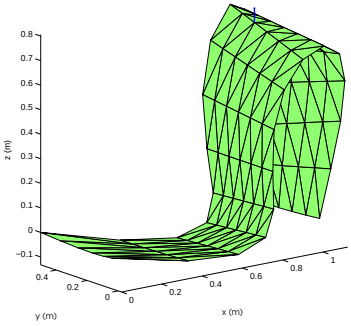


Figure B-9: Load case 8: setup

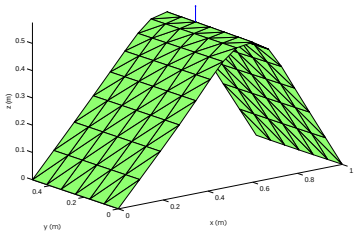
As a final conclusion it can be said that the shell element implementation was successful and works as supposed and is suited for the use in the canopy modelling.



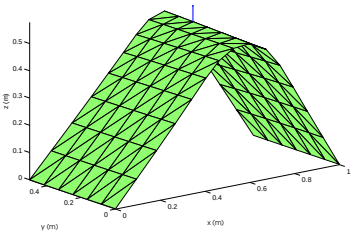
(a) Linear, MATLAB , deformation scale = 1, maximum deformation = 0.7203m



(b) Linear, Ansys, deformation scale = 1, maximum deformation = 0.7310m



(c) Non-linear, MATLAB , deformation scale = 1, maximum deformation = 0.0808m



(d) Non-linear, Ansys, deformation scale = 1, maximum deformation = 0.0875m

Figure B-10: The results for load case 8 for the shell element.

B-3 Beam elements

B-3-1 Set-up

For the test with beam elements a steel beam with a length of 1 meter and a circular diameter of 0.05m was taken. The used steel properties are $E = 200Gpa$ and $\nu = 0.3$. The beam is meshed with 20 elements and therefore 21 nodes with a total of 132 degrees of freedom.

In MATLAB the beam will be modelled with the elements as described in section A. In Ansys the BEAM4 element type has been chosen, since this one comes the closest to the developed MATLAB beam element, because it is based on the Bernoulli-Euler beam theory as well. An exact agreement is not to be expected since the BEAM4 element uses some different theories.

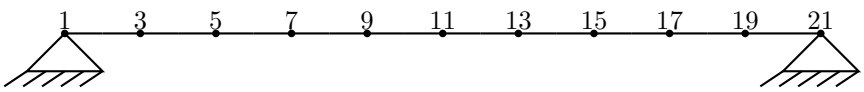


Figure B-11: Load case 1: setup

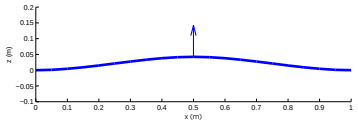
B-3-2 Load cases

Load case 1

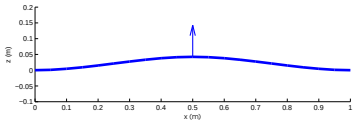
Load case: as in Figure B-11.
Boundary conditions: clamped at end nodes.
Load: $F_z = 10000N$ for node 11.
Results: Table B-9 and Figure B-12.

Table B-9: Results load case 1 beam element

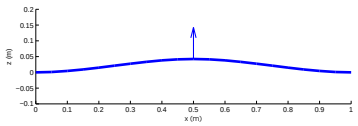
	MATLAB	Ansys	Error
Linear	8.488×10^{-4}m	8.488×10^{-4}m	2.37×10^{-6}
Non-linear	8.486×10^{-4}m	8.488×10^{-4}m	1.22×10^{-6}



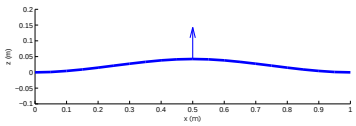
(a) Linear, MATLAB , deformation scale = 50, maximum deformation = 8.488×10^{-4}m



(b) Linear, Ansys, deformation scale = 50, maximum deformation = 8.488×10^{-4}m



(c) Non-linear, MATLAB , deformation scale = 50, maximum deformation = 8.486×10^{-4}m



(d) Non-linear, Ansys, deformation scale = 50, maximum deformation = 8.488×10^{-4}m

Figure B-12: The results for load case 1 for the beam element.

Load case 2

Load case: same as load case 1, except for the force direction.
Boundary conditions: clamped at end nodes.
Load: $F_y = 10000N$ for node 11.
Results: Table B-10.

Table B-10: Results load case 2 beam element

	MATLAB	Ansys	Error
Linear	8.488×10^{-4}m	8.481×10^{-4}m	1.20×10^{-4}
Non-linear	8.486×10^{-4}m	8.481×10^{-4}m	1.22×10^{-6}

Load case 3

Load case: same as load case 1, but applying a torsion moment instead of a force.
Boundary conditions: clamped at end nodes.
Load: $M_x = 10000N$ for node 11.
Results:Table B-11.

Table B-11: Results load case 3 beam element

	MATLAB	Ansys	Error
Linear	0.0530deg	0.0530deg	6.59×10^{-7}
Non-linear	0.0530deg	0.0530deg	6.59×10^{-7}

Load case 4

Load case: same as load case 1, except for the force direction.
Boundary conditions: clamped at end nodes.
Load: $F_y = 1MN$,$F_z = 1MN$ for node 11.
Results: Table B-12.

Table B-12: Results load case 4 beam element

	MATLAB	Ansys	Error
Linear	0.120m	0.120m	4.73×10^{-6}
Non-linear	0.0556m	0.0561m	1.70×10^{-3}

Load case 5

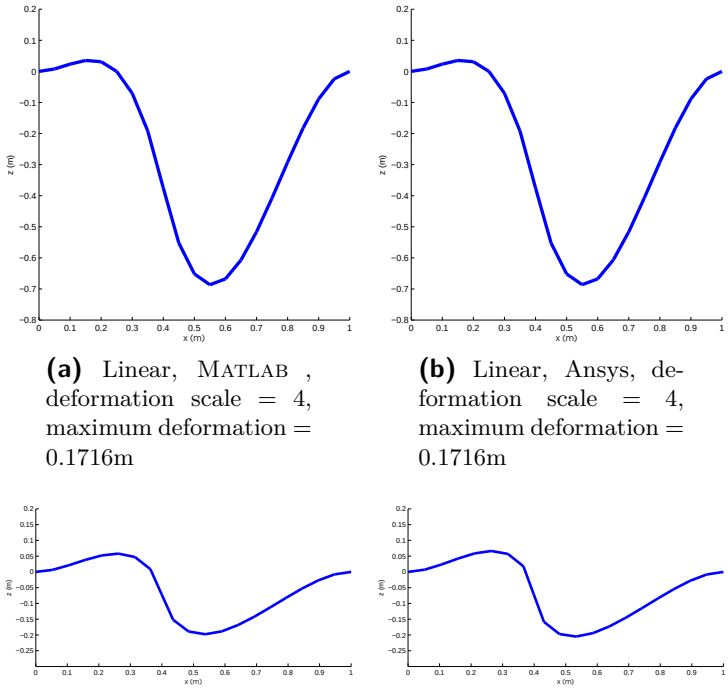
Load case: same as load case 1, except for the force direction and application point.
Boundary conditions: clamped at end nodes.
Load: $M_y = 1MNm$ for node 9.
Results: Table B-13 and Figure B-13.

Table B-13: Results load case 5 beam element

	MATLAB	Ansys	Error
Linear	0.1716m	0.1716m	9.89×10^{-7}
Non-linear	0.0495m	0.0514m	0.0033

Load case 6

Load case: same as load case 1.
Boundary conditions: clamped at the left node.
Load: $F_z = -30.000N$ for node 21.
Results: Table B-14 and Figure B-14.



(a) Linear, MATLAB , deformation scale = 4, maximum deformation = 0.1716m

(b) Linear, Ansys, deformation scale = 4, maximum deformation = 0.1716m

(c) Non-linear, MATLAB , deformation scale = 4, maximum deformation = 0.0495m

(d) Non-linear, Ansys, deformation scale = 4, maximum deformation = 0.0514m

Figure B-13: The results for load case 5 for the beam element.

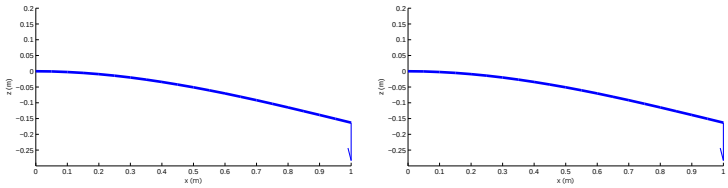
Table B-14: Results load case 6 beam element

	MATLAB	Ansys	Error
Linear	0.1630m	0.1630m	2.52×10^{-6}
Non-linear	0.1584m	0.1595m	0.0018

B-3-3 Conclusions

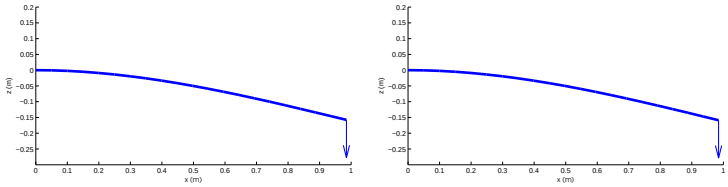
The following can be seen from these load cases:

- In general the results for the MATLAB and Ansys model are very similar, which leads to the conclusion that the developed beam model is accurate enough for the purpose of this research.
- The convergence of the MATLAB element is very fast.
- The element is slightly stiffer than the Ansys element.



(a) Linear, MATLAB , deformation scale = 1, maximum deformation = 0.1630m

(b) Linear, Ansys, deformation scale = 1, maximum deformation = 0.1630m



(c) Non-linear, MATLAB , deformation scale = 1, maximum deformation = 0.1584m

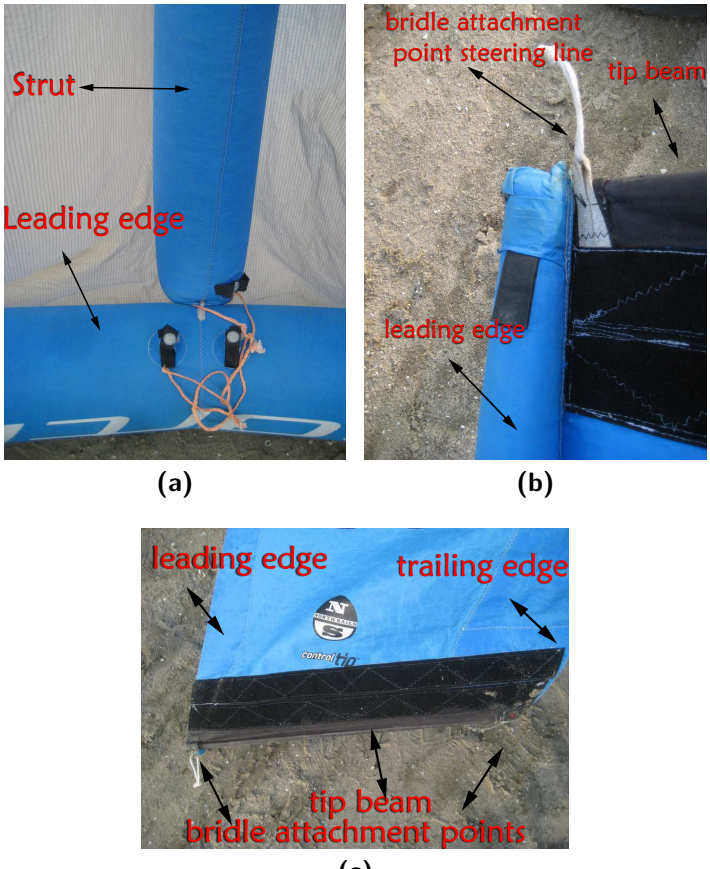
(d) Non-linear, Ansys, deformation scale = 1, maximum deformation = 0.1595m

Figure B-14: The results for load case 6 for the beam element.

Appendix C

North Rhino 16

C-1 Detailed photos



(a) (b)

(c)

Figure C-1: Details of the North Rhino 16

(a) **(b)**

Figure C-2: The North Rhino 16

Figure C-3: Deformations while flying.

C-2 Geometric properties

Table C-1: Geometric properties of the kite

Parameter	Value
L_{strut0}	1.93 m
L_{strut1}	1.67 m
L_{strut2}	1.08 m
L_{tip}	0.65 m
$r_{LE,mid}$	100 mm
$r_{LE,tip}$	40 mm
$r_{strut0,LE}$	50 mm
$r_{strut0,TE}$	35 mm
$r_{strut1,LE}$	40 mm
$r_{strut1,TE}$	35 mm
$r_{strut2,LE}$	35 mm
$r_{strut2,TE}$	35 mm
Angle strut0	0 deg
Angle strut1	36.5 deg
Angle strut2	72 deg
Height	3.20 m
Span	5.80 m

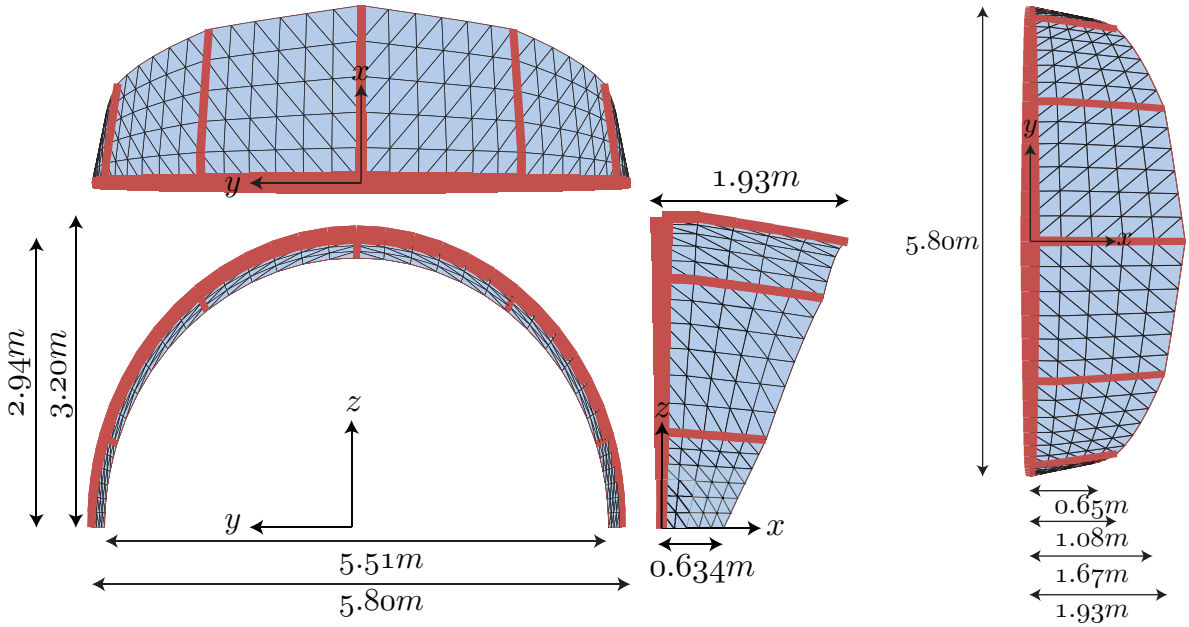


Figure C-4: The dimensions of the North Rhino $16m^2$ kite.

C-3 Deformation sequence figure eight flight

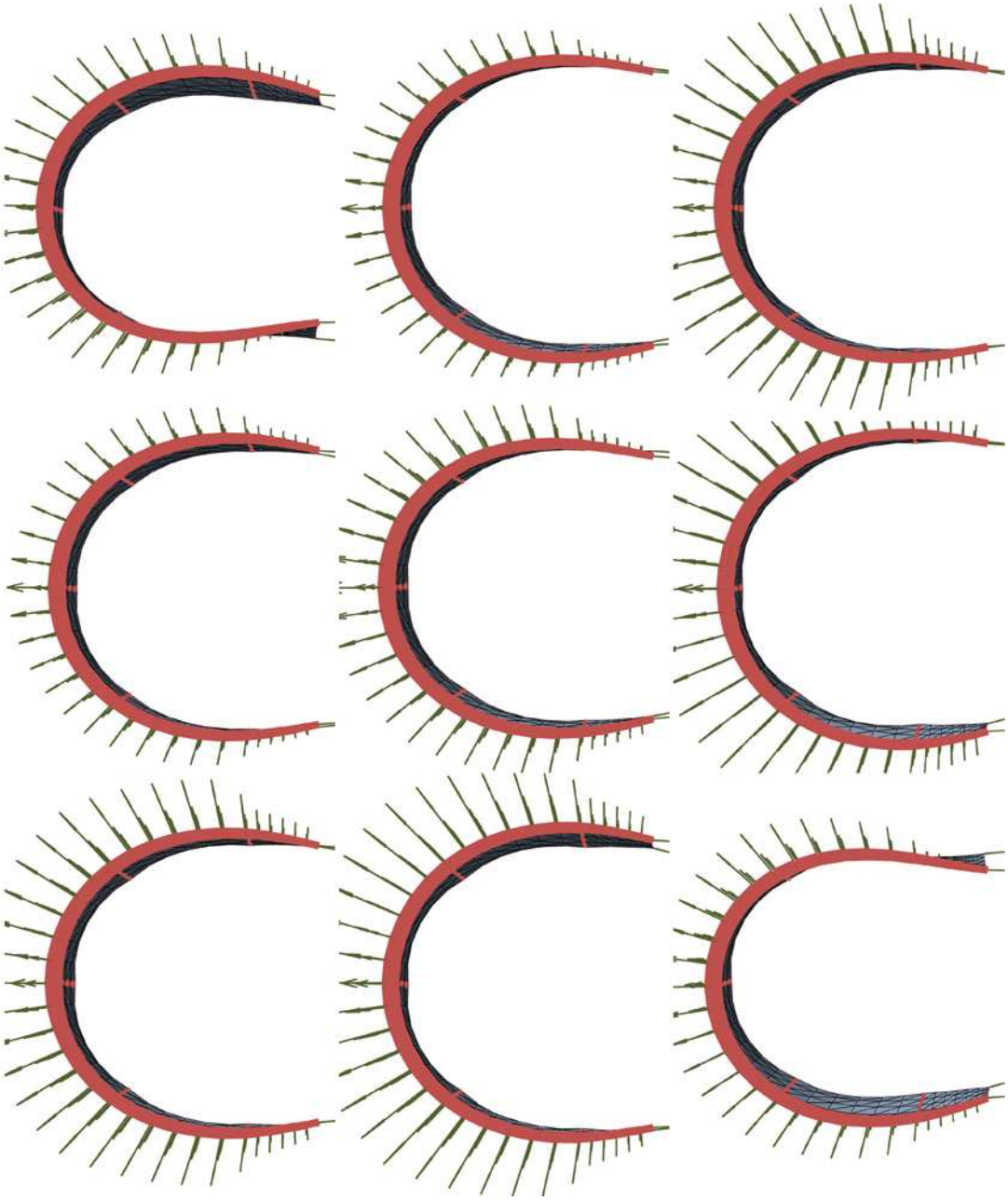


Figure C-5: Deformation (amplified by a factor 2) of the kite while flying a figure eight.

Bibliography

[1] International energy outlook 2011. Technical report, U.S. Energy Information Administration, 2011.

[2] John F. Abel and Mark S. Shephard. An algorithm for multipoint constraints in finite element analysis. *International Journal for Numerical Methods in Engineering*, 14(3): 464–467, 1979. ISSN 1097-0207.

[3] D.J. Allman. A simple cubic displacement element for plate bending. In *International Journal for Numerical Methods in Engineering*, volume 10, pages 263–281. 1976.

[4] D.J. Allman. Evaluation of the constant strain triangle with drilling rotations. In *International Journal for Numerical Methods in Engineering*, volume 26, pages 2645–2655. 1988.

[5] J.H. Baayen. Modeling a kite on a variable length flexible tether. Master's thesis, Delft University of Technology, 2011.

[6] B.Fraeijs and De Veubeke. The dynamics of flexible bodies. *International Journal of Engineering Science*, 14(10):895 – 913, 1976.

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

[8] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current*, 78(7):808–817, 2000.

[9] Saifon Chaturantabut and Danny C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[10] M.A. Crisfield. *Non-linear Finite Element Analysis of Solids and Structures, Volume 1: Essentials*. John Wiley and Sons, 1991.

[11] S.G.C. de Groot. Modelling the dynamics of an arc-shaped kite for control law design. Master's thesis, Delft University of Technology, 2010.

[12] A. de Wachter. Power from the skies - laddermill takes airborne wind energy to new height. *Leonardo Times - Journal of the Society of Aerospace Engineering Students 'Leonardo da Vinci'*, (4):18–20, 2010.

[13] Michael Erhard and Hans Strauch. Control of towing kites for seagoing vessels. *CoRR*, abs/1202.3641, 2012.

[14] C. Felippa. *Nonlinear Finite Element Methods*. University of Colorado at Boulder, 2010.

[15] C.A. Felippa. A systematic approach to the element-independent corotational dynamics of finite elements. Technical Report Technical Report CU-CAS-00-03, Center for Aerospace Structures - College of Engineering - University of Colorado, 2000.

[16] C.A. Felippa. A study of optimal membrane triangles with drilling freedoms. Technical Report Technical Report CU-CAS-03-02, Center for Aerospace Structures - College of Engineering - University of Colorado, 2003.

[17] Allister Furey and Inman Harvey. Evolution of neural networks for active control of tethered airfoils. *Neural Networks*, 4648 LNAI:746–755, 2007.

[18] Inman Harvey and Allister Furey. Adaptive behavioural modulation and hysteresis in an analogue of a kite control task. *From Animals to Animats 10*, 5040:509–518, 2008.

[19] R.C. Hibbeler. *Mechanics of materials*. Prentice Hall, 2003.

[20] S.F. Hoerner. Fluid dynamic drag. *USA: Hoerner Fluid Dynamics*, 1965.

[21] B. Houska. A 9 dof kite model. Internal Paper, University of Heidelberg.

[22] David V. Hutton. *Fundamentals of finite element analysis*. Mc Graw Hill, 2004.

[23] Claudius Jehle. Controller design for kite powered systems, wip. Master's thesis, Delft University of Technology, 2012.

[24] Libor Jendele and Jan Cervenka. On the solution of multi-point constraints application to fe analysis of reinforced concrete structures. *Computers and Structures*, 87(15):970 – 980, 2009. ISSN 0045-7949.

[25] P Krysl, S Lall, and J E Marsden. Dimensional model reduction in nonlinear finite element dynamics of solids and structures. *International Journal for Numerical Methods in Engineering*, 51(4):479–504, 2001.

[26] S Lall. Structure-preserving model reduction for mechanical systems. *Physica D: Nonlinear Phenomena*, 184(1-4):304–318, 2003.

[27] B. Lansdorp, R. Ruiterkamp, and W. Ockels. *Towards flight testing of remotely controlled surfkites for wind energy generation*, volume 6643. 2007.

[28] Bas Lansdorp and W.J. Ockels. Design of a 100 mw laddermill for wind energy generation from 5 km altitude 2. *Design*, pages 5–10, 2005.

[29] Bas Lansdorp and Paul Williams. The laddermill - innovative wind energy from high altitudes in holland and australia. pages 1–14.

[30] Bas Lansdorp, B. Remes, and W.J. Ockels. Design and testing of a remotely controlled surfkite for the laddermill. *Wind Energy*, pages 1–4, 2005.

[31] Bas Lansdorp, Richard Ruiterkamp, Paul Williams, and Wubbo Ockels. Long-term laddermill modeling for site selection. *Time*, (August), 2008.

[32] Miles L. Loyd. Crosswind kite power. *Journal of Energy*, 4(3), May-June 1980.

[33] Empa materials science and technology. Biaxial test report ripstop nylon. Technical report, Empa, 2011.

[34] J.A. Mulder, W.H.J.J. van Staveren, and E. de Weerdt J.C. van der Vaart. *Flight Dynamics*. Delft University of Technology, 2006. Flight Dynamics Course.

[35] O. S. Narayanaswamy. Processing nonlinear multipoint constraints in the finite element method. *International Journal for Numerical Methods in Engineering*, 21(7):1283–1288, 1985. ISSN 1097-0207.

[36] W.J. Ockels. Kite power and propulsion, the laddermill principle. *Submitted for publication*, 2009. ASSET Delft University of Technology.

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

[38] Muruhan Rathinam and Linda R Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893, 2003.

[39] J.N. Reddy. *Nonlinear Finite Element Analysis*. OXFORD University Press, 2004.

[40] Daniel J. Rixen. *Numerical methods in Engineering Dynamics*. Delft University of Technology, 2007.

[41] A L Schwab. Detc2006-99307 how to draw euler angles and utilize euler parameters. *Engineering Conference*, 2006.

[42] A.L. Schwab. *Multibody Dynamics B*. Delft University of Technology, 1998.

[43] Joost Schwoll. Finite element approach for statically loaded inflatable kite structures,wip. Master's thesis.

[44] Ahmed A. Shabana, Olivier A. Bauchau, and Gregory M. Hulbert. Integration of large deformation finite element and multibody system algorithms. *Journal of Computational and Nonlinear Dynamics*, 2(4):351–359, 2007.

[45] G.J. Spierenburg. Continued development and experimental validation of a kite design and simulation tool. Master's thesis, Delft University of Technology, 2005.

[46] E.J. Terink. Kiteplane flight dynamics. Master's thesis, Delft University of Technology, 2009.

[47] Paolo Tiso. *Finite element based reduction methods for static and dynamic analysis of thin-walled structures*. PhD thesis, University of Technology Delft, 2006.

[48] John van den Heuvel. Kite sailing - improving system performance and safety. Master's thesis, Delft University of Technology, 2010.

[49] S.L. Veldman, O.K. Bergsma, and A. Beukers. Bending of anisotropic inflated cylindrical beams. *Thin-Walled Structures*, 43(3):461 – 475, 2005. ISSN 0263-8231.

[50] S Volkwein. Model reduction using proper orthogonal decomposition. *Lecture Notes Institute of Mathematics and Scientific Computing University of Graz*, pages 1–42, 2008.

[51] Jinzhong Wang, Zheng-Dong Ma, and Gregory M. Hulbert. A distributed mechanical system simulation platform based on a gluing algorithm'. *Journal of Computing and Information Science in Engineering*, 5(1):71–76, 2005.

[52] Paul Williams. *Optimal Wind Power Extraction with a Tethered Kite*, volume 2, pages 1167–1195. 2006.

[53] Paul Williams, Bas Lansdorp, and Wubbo Ockels. Flexible tethered kite with moveable attachment points, part i: Dynamics and control. *Dynamics and Control*, (August): –6628, 2007.

[54] Paul Williams, Bas Lansdorp, and Wubbo Ockels. Modeling and control of a kite on a variable length flexible inelastic tether. *AIAA Guidance navigation and control conference*, (August):–6705, 2007.

[55] Paul Williams, Bas Lansdorp, and Wubbo Ockels. Optimal crosswind towing and power generation with tethered kites. *Journal of Guidance, Control, and Dynamics*, 31(1): 81–93, January 2008.

[56] Paul Williams, Bas Lansdorp, Richard Ruiterkamp, and Wubbo Ockels. Modeling, simulation, and testing of surf kites for power generation. *Aerospace*, (August), 2008.