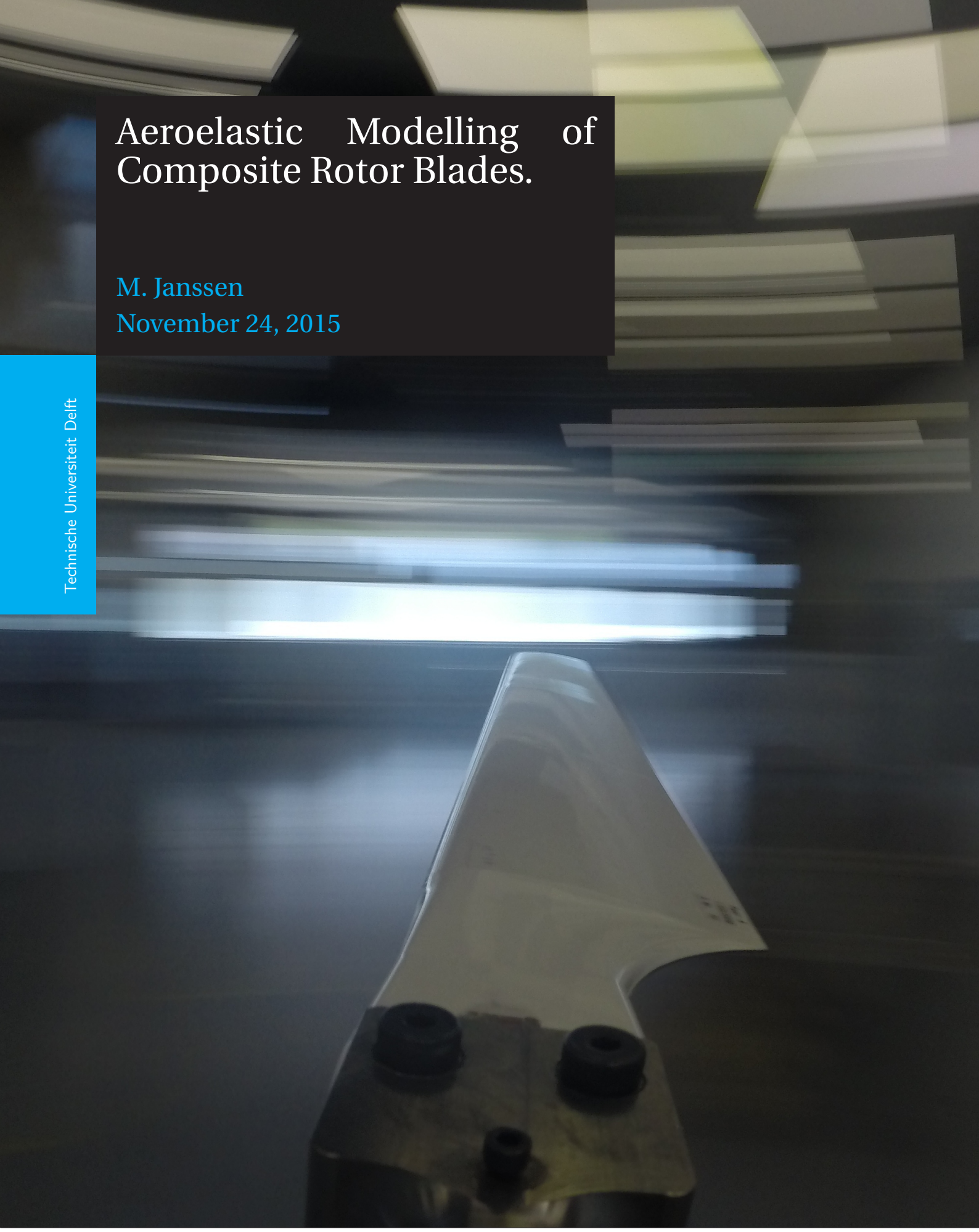
Aeroelastic Modelling of Composite Rotor Blades.

M. Janssen

November 24, 2015

Technische Universiteit Delft



TU Delft Delft University of Technology

Challenge the future

Aeroelastic Modelling of Composite Rotor Blades.

by

M. Janssen

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace and Space Engineering

at the Delft University of Technology,
to be defended publicly on Friday December 11, 2015 at 13:00.

Supervisor:	Dr.ir. M. Voskuijl	Delft University of Technology
Thesis committee:	Prof.dr.ir. L.L.M. Veldhuis	Delft University of Technology
	Dr.ir. M. Voskuijl	Delft University of Technology
	Dr.ir. M. D. Pavel	Delft University of Technology
	Ir. A. A. Jongbloed	KVE Composites Group

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Thesis registration number: 060#15#MT#FPP.

TU Delft Delft University of Technology

ABSTRACT

This thesis was conducted at KVE Composites Group (The Hague, The Netherlands). The aim of this thesis is to create an aeroelastic model that can be used during the design phase of these rotor blade projects.

An aeroelastic was created consisting of a Ritz-Hamilton structural model, a lifting line quasi steady aerodynamic model with two inflow models. The aeroelastic model includes effects due to centripetal and gravity forces and conservation of angular momentum. A fourth order Runge-Kutta integration method is used to numerically integrate the aeroelastic model to obtain the response of the system.

The aeroelastic model was validated by comparing results calculated using the aeroelastic model with theoretical results and results obtained by measurements. A uniform beam and several rotor blades were modelled for this validation.

The calculated steady state deflections of the uniform beam were compared with values obtained from literature. This comparison showed the results of the aeroelastic model match the results from literature, which indicates (the steady state part of) the structural model works correctly.

The calculated steady state deflections of the rotor blades were compared with values obtained from measurements on the rotor blades. This comparison showed that there is some error between the results from the aeroelastic model and the measurements. For the aeroelastic model using 5 elements the error is relatively large and for the aeroelastic model using 15 elements the error is greatly reduced.

The calculated natural frequencies for a uniform beam were compared to values obtained from literature. This comparison showed that with only 8 elements an accurate prediction can be made of the first two natural frequencies for the torsion, longitudinal, flap and lead-lag motions. The predictions of the third natural frequencies for the flap and lead-lag motion were also very accurate, while the prediction for the longitudinal and torsion motion required a few more elements.

The calculated natural frequencies for a rotor blade were compared with values obtained by measurements performed by the National Aerospace Laboratory (NLR). This comparison showed that the aeroelastic model using only 5 elements is not capable of predicting the natural frequencies of the rotor blade accurately. However, if an aeroelastic model with 15 elements is used the calculated natural frequency for the first torsion mode, the first two flap bending modes and the first lead-lag bending mode match the measured natural frequencies very well. For the third bending flap bending mode to be accurate an aeroelastic model using more elements is required.

The calculated drag for the rotor blades was compared with values obtained in a series of whirl tower tests. The power curves predicted by the aeroelastic model have a similar shape as power curves created using the whirl tower data. The power required predicted by the aeroelastic model at low collective angles is slightly higher than the power required values measured. The predicted lift at high collective angles is slightly higher than can be expected based on the maximum take of weight and performance of the helicopter.

Using this model KVE is able to predict initial loads, deflections and the consequences of changes in the design of their rotor blade.

ACKNOWLEDGEMENTS

Foremost my most sincere gratitude goes to my supervisors Auke Jongbloed, Mark Voskuijl, Marilena Pavel and Christos Kassapoglou for their advice, guidance and insights.

I would also like to thank KVE Composites Group for the opportunity to do two great projects. I have enjoyed working on both projects and learned a lot from everyone. Thank you for providing the rotor blades, equipment and resources to perform al the measurements.

Finally I would like to thank Cynthia Nijenhuis for her patience and support.

M. Janssen
November 2015

CONTENTS

LIST OF FIGURES

NOMENCLATURE

α_{ampl}	Cyclic amplitude	$[^\circ deg]$
Ψ	Azimuth angle	$[rad]$
α	Angle of attack	$[^\circ deg]$
α_0	Angle of incidence of the root section	$[^\circ deg]$
α_{coll}	Collective angle	$[^\circ deg]$
β	Flapping angle	$[Rad]$
η	Lead-lag angle	$[Rad]$
Ω	Rotational speed	$[Rad/s]$
ρ	Density	$[kg/m^3]$
θ	Angle of incidence	$[^\circ deg]$
θ_1	Angle caused by movement of the blade	$[^\circ deg]$
θ_2	Angle caused by movement of the flow	$[^\circ deg]$

A	Cross sectional area	$[m^2]$
AR	Aspect ratio	$[-]$
c	Chord length	$[m]$
C_D	Drag coefficient of a wing	$[-]$
C_d	Drag coefficient of an airfoil	$[-]$
C_l	Lift coefficient of an airfoil	$[-]$
C_m	Aerodynamic moment coefficient of an airfoil (2d)	$[-]$
C_m	Aerodynamic moment coefficient of an airfoil	$[-]$
$C_{L_{vi=0}}$	Lift coefficient of a wing, neglecting the induced velocity of the flow	$[-]$
D	Drag	$[N]$
E	Modus of elasticity	$[Pa]$
e	Oswald factor	$[-]$
F	Forces and moments matrix	$[-]$
f_x	Force in x direction	$[N]$
G	Modulus of rigidity	$[Pa]$
I	Polar moment of area	$[m^4]$
K	Stiffness matrix	$[-]$

L	Lift	$[N]$
l	Length	$[m]$
M	Inertia matrix	$[-]$
M	Moment	$[Nm]$
R	Maximum radius	$[m]$
r	Radius, distance from rotational centre	$[m]$
S	Wing area	$[m^2]$
u	Displacement vector	$[-]$
x	Displacement or position in span-wise direction	$[m]$
y	Displacement or position in leading edge-trailing edge direction	$[m]$

1

INTRODUCTION

This thesis was conducted at KVE Composites Group (The Hague, The Netherlands). KVE designs, manufactures and repairs high-performance composite products for aeronautical industry, defence industry, medical industry and mechatronics. One of the company's specialities is the development and production of composite rotor blades for the large unmanned helicopters. The aim of this thesis is to create an aeroelastic model that can be used during the design phase of these rotor blade projects.

The focus of most aeroelasticity research is aimed on improving accuracy and reducing computational effort [2–7]. However for this thesis the usability by people that are not aeroelastic experts is also important. Schuster et al described aeroelastic results obtained in the past, the progress that has been made and challenges for the future of aeroelasticity[4]. They show examples of successful computational flutter analysis, but also provide the following advice:

"Perhaps the most daunting challenge faced by developers of future aeroelasticity analysis and design methods is the formulation of tools that can be employed by engineers who are not experts in the individual disciplines that form the methods."

This thesis adds practical value to the existing models explaining aeroelastic behaviour of composite rotor blades. The model described in here is a practical translation of scientific literature and can be used as a hands-on tool in composite rotor blade testing and development.

The objective of this thesis is described in the following main question:

How can aeroelastic behaviour of a composite rotor blade be modelled, for the purpose and accuracy desired?

This question consists of two parts: the modelling part and the purpose and accuracy part. A third aspect is the verification whether the desired purpose and accuracy are met. This results in three sub-questions:

What is the desired purpose and accuracy?
How to create an aeroelastic model?
How to verify the accuracy of the model?

The first sub-question is answered in chapter 2. This chapter describes KVE's development process and the purpose of the aeroelastic model in this process and the accuracy required to serve this purpose.

This second sub-question is answered in chapter 3. This chapter describes the theory of aeroelasticity and the implementation of this theory to create the aeroelastic model.

The third sub-question is answered is chapters 4 and 5. KVE provided several rotor blades from one of their rotor blade projects and equipment to validate (parts of) the aeroelastic model. These rotor blades and

equipment is described in chapter 4. The result of this validation can be found in chapter 5.

Finally chapter 6 contains the conclusions and recommendations.

2

BACKGROUND

This chapter describes the process of rotor blade development by KVE and the intended role of the aeroelastic model in the development process for future projects. In addition the requirements for the aeroelastic model in order to full fill this role are described.

2.1. DEVELOPMENT PROCESS

KVE's process of development of rotor blades is summarized below in a list of steps. The actual development process is more complex than can be captured by a list of steps. Adjustments in the structural design can be made at any point. The development of rotor composite rotor blades consists of the following steps:

- Listing of the specifications
- Structural design
- Stability and (dynamic) stresses calculations
- Production of a prototype
- Static and dynamic testing of the prototype

2.1.1. LISTING OF SPECIFICATIONS

KVE's specialities are designing and manufacturing of composite components. Aerodynamic design of rotor blades is not an area of expertise of KVE. Therefore the shape of the rotor blade is usually specified by the customer. The intended operational rotation speed is related to this aerodynamic design and is provided often as well. Other specifications can be different for each project, but can include stability criteria (stable behaviour at certain rpm range, natural frequency not near rpm range), weight or centre of gravity position criteria, stiffness or maximal deflection under specified conditions criteria.

2.1.2. STRUCTURAL DESIGN

The structural design and stability and stresses calculations are less a separate processes then the list of steps suggests. There is some overlap and some feedback between these to steps. Before the structural design is started simple loads estimates are made. Examples of these simplified estimates are constant lift over the length of the rotor blade, with the total lift equal to the maximum total lift. Another example is calculating the centripetal force at the root of the blade by representing the rotor blade as a point mass. The mass and position of this point mass are set equal to the mass of the rotor blade and position of the centre centre of gravity of the the rotor blade. These loads are compared to the loads from previous rotor blade projects and a structural design is made.

2.1.3. STABILITY AND STRESSES CALCULATIONS

This structural design is used for stability and stress analysis. The static stresses calculation is done by KVE using ANSYS software. The estimated loads are used to calculate the resulting stresses in the rotor blade. Depending on the outcome the structural model is adjusted and the stresses recalculated. If the structural

design of the rotor blade has passed this static stress analysis the dynamic loads and stability calculations are started. These dynamic loads and stability calculations are performed by the National Aerospace Laboratory (NLR). The goal of these calculations is to investigate whether adjustments of the structural design are required to obtain a rotor blade with a stable behaviour for the operational range of rotational speeds. The dynamic loads can be compared to the estimated loads used for the structural design. If the dynamic loads are significant larger than the estimated loads the static stress analysis can be repeated.

2.1.4. PRODUCTION OF A PROTOTYPE

If the rotor blade design passes both the static stress analysis and the stability calculations the production of a prototype can be started. The number of prototypes produced depends on the number and (destructvie) nature of the tests that will be performed with these prototypes. Usually one or two prototypes are made at a time.

2.1.5. STATIC AND DYNAMIC TESTING

After production the prototype is tested to verify if the rotor blade is within the specifications stated in the first step. The tests that are performed depend on the specifications and can be different for each project. Examples of tests that were performed previously are a pull test, a stiffness measurement, a weight of centre of gravity measurement, and a whirl tower test.

A pull test was performed for the blades described in chapter 4 at the Delft University of Technology. The blade was fixed at the root. A force was applied half way between the root and the tip. This force was increased until the rotor blade collapsed. This test showed whether the root of the prototype was able to handle the large centripetal forces.

If a stiffness requirement or maximum deflection requirement is provided the stiffness of the prototype is measured. The process of this stiffness measurement is described in section 4.2. A picture of determining the stiffness of a rotor blade is shown in figure 4.2.

Usually there is a weight requirement or bounds for the position of the centre of gravity of the rotor blade specified. The process of measuring weight and determining the centre of gravity is described in section 4.3. A picture of determining the weight and position of the centre of gravity is shown in figure 4.6.

If all tests show that the rotor blade is within specifications a dynamic test can be performed. This can be done by fitting a set of blades on the actual helicopter. However using the actual helicopter is is expensive and the consequences of a rotor blade failure would eben more expensive. Alternatively one blade, or a set of blades can be tested on the whirl tower.

WHIRL TOWER

The development of the KVE whirl tower started in October 2012. It was the authors internship assignment to design and build it. The whirl tower was completed at the end of January 2013 and it was operational for testing one month later. At this point KVE and the Delft University of Technology are in search for a permanent location for the whirl tower.

The whirl tower consist of a frame, an electric motor, an axle, a swashplate and a V-link transmitter. The whirl tower can be used to spin either one rotor blade and a counterweight or two rotor blades up to speeds of 1400rpm at a constant or cyclic angle of incidence. The V-link is an wireless 7 channel input sensor node and can be used to transmit data from sensors, such as strain gauges, piezo sensors or angle measurements. The whirl tower is powered by a 75kW electric motor controlled by an programmable inverter. The inverter can measure the voltage and current delivered to the motor and the motors rotational speed. A picture from the top of the whirl tower, while spinning a blade, can be seen on the cover of this report. Two other pictures of the whirl tower in action is shown in figures 2.1 and 5.10.

Several tests can be performed using the whirl tower, for example an dynamic deflection test or measuring the power required to rotate a blade. For the blades described in chapter 4 an endurance test was performed. A prototype of the rotor blade was spun for 30 hours at operational rotational velocity and at a cyclic angle of incidence. After this endurance test another pull test was performed. The results of this pull tests were

Figure 2.1: KVE's whirl tower in action

compared to the results of a previously performed pull test (with another, new rotor blade), to determine the wear and damage caused by the 30 hours on the whirl tower.

2.2. PURPOSE AND ROLE OF THE AEROELASTIC MODEL

The aim of this thesis is to provide KVE a tool for the third step: the stability and (dynamic) stresses calculations. This part was performed by the NLR. However KVE would like to be able to do this in-house for future projects.

The aim of this project is to create an aeroelastic model which enables KVE to:
- Determine the static and dynamic deformations of a rotor blade during a specified manoeuvre.
- Determine the static and dynamic forces acting on a rotor blade during a specified manoeuvre.
- Determine the natural frequencies of the rotor blade at specified settings (rotational speed, pitch, etc.).
- Evaluate the consequences of design changes on the stability, deformations and stresses.
- Compare the aeroelastic behaviour of different rotor blade designs.

A specified manoeuvre can be hoover (close to the the ground or not), forward flight, climb etc. Which manoeuvre is specified depends on the specifications stated in the first step of the development process. For example, a specification could be: Maximum flap deflection of 10cm or less at maximum speed forward flight. The forces acting on the rotor blade depend on the specified manoeuvre. This means this specified manoeuvre needs to be part of the input of the aeroelastic model.

The deformations and forces are important parameters for the structural design. The deformations of the rotor blade or the forces acting on the rotor blade can be translated to stresses. This can be done using ANSYS or an other program. The forces obtained can be compared to the previously estimated forces. This allows KVE to determine stress concentrations and optimize the structural design. The deformations are also important if listed in the specifications, like the example given in the specified manoeuvre section.

The natural frequencies of the rotor blades are important parameters for the stability of the helicopter. If any of the natural frequencies match or are near the operation frequency of the helicopter resonance may occur. This can cause parts to break and should be avoided if possible.

KVE would like the use the aeroelastic model to evaluate the consequences of design changes. This is necessary if a rotor blade is not performing as expected or as required. Making adjustments in the structural design, adjusting the input files of the structural model is far less costly in terms of time and money compared to making adjustments in the structural design, producing a prototype and test it on the whirl tower.

Comparing the aeroelastic behaviour of different rotor blades is very useful if KVE is asked to build a replacement for an existing rotor blade. This was the case for example for the rotor blades described in chapter 4.

KVE did not specify the required accuracy for the determination of strains, stressed and the natural frequencies. However an estimate for the required accuracy based on the purpose of the aeroelastic model and KVE's development process described in this sections was made: The loads and strains should be predicted within a 20% and natural frequencies within a 5% margin of accuracy.

3

MODELING

This chapter describes the theory and implementation of the aeroelastic model. The base of an aeroelastic model consist of a structural model (inertias and structural forces), and external forces [8, 9]. The external forces can be split into aerodynamic forces and other than aerodynamic forces.

The first section of this chapter describes the structural model, the second section describes the aerodynamic model. The third section describes forces other than structural and aerodynamic forces and the final section describes the process of combining these aspects into an aeroelastic model and use it to obtain results. An overview of the files and functions developed and used during this thesis can be found in appendix B.

3.1. STRUCTURAL MODEL

A structural model describes the interaction between the elastic energy due to deformation of the structure and the kinetic energy due to motion of the structure [8, 9]. These two factors determine the motion of a free vibrating beam, wing or rotor blade. In the first subsection of this section the selection of a structural model is explained. In the second subsection the theory of the method of choice is described. In the final subsection the implementation of the theory into code is described.

3.1.1. MODEL SELECTION

Several models were considered. The first and most simple model is the typical section model. Increasing in complexity, the next model is the Lumped Parameter model, a much used model. The third model is based on Rayleigh's Principle, the fourth will be an improved variation called the Ritz-Hamilton method. The fifth model is the Goland Beam model and finally finite element methods will be discussed, briefly.

The typical section method is a very simple model. It has only two degrees of freedom and it can not simulate the behaviour of a complete wing or rotor blade correctly. The lumped parameter method is more accurate. However obtaining the input parameters might be difficult. Rayleigh's method is suitable, but only works for known mode shapes. Determining these mode shapes increase its complexity. The Ritz-Hamilton method is more accurate then the Rayleigh's method. Furthermore its accuracy can be improved by using more elements, allowing it's accuracy to be tailored to requirements. Like the lumped parameter method it does requires the determination of input variables. These variables are properties like mass and stiffness. These properties are measured to determine if the blades are within specifications, regardless of the choice of structural model. The Goland beam is more limited compared to the Ritz method. It has only two degrees of freedom and is therefore not suitable for this application. It should be possible to obtain accurate results using an off the shelf FEM package. Static loads can be relatively easy, but dynamic loads might require more effort to implement. However a structural CAD model is required, but not always available. Therefore the Ritz method is the most suitable model for this project, given the time and resources available.

3.1.2. DESCRIPTION

The Ritz-Hamilton method and the derivation of its equations is described in many books and papers [8–15]. The Ritz-Hamilton structural model divides a beam into a number of elements. The position of one element is described by two sets of coordinates or nodes. One node is positioned at the beginning of an element and one at the end of an element. Two adjacent elements share the node between them, so a model that consists of n elements is described by $n+1$ nodes. This is shown in figure 3.1. Displacements and loads are transferred through this shared node from one element to the other. Using a summation of forces an expression for the displacement can be derived. This can be used in expressions for kinetic and strain energy. By substituting these in Lagrange's equation an expression for the motion can be derived.

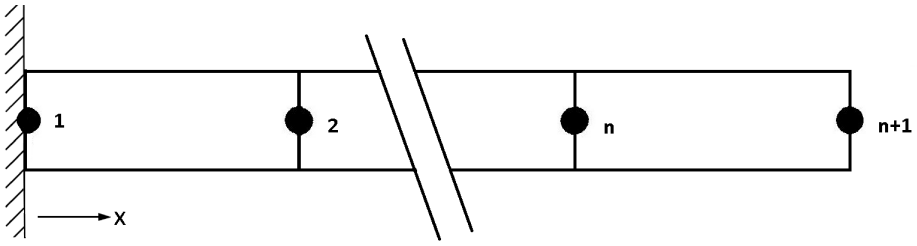


Figure 3.1: A beam divided into n elements.

The set of equations for torsional motion for a single element i is given by:

$$\frac{\rho_i I_i l_i}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{bmatrix}\theta_i''\\ \theta_{i+1}''\end{bmatrix}+\frac{9G_i I_i}{l_i}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\begin{bmatrix}\theta_i\\ \theta_{i+1}\end{bmatrix}=\begin{bmatrix}M_{xi}\\ M_{xi+1}\end{bmatrix} \tag{3.1}$$

The set of equations for longitudinal vibration for a single element i is given by:

$$\frac{\rho_i A_i l_i}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix}\begin{bmatrix}x_i''\\ x_{i+1}''\end{bmatrix}+\frac{9E_i A_i}{l_i}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\begin{bmatrix}x_i\\ x_{i+1}\end{bmatrix}=\begin{bmatrix}f_{xi}\\ f_{xi+1}\end{bmatrix} \tag{3.2}$$

The sets of equations for bending motion for a single element i is given by:

$$\frac{\rho_i A_i l_i}{420}\begin{bmatrix}156 & 22l_i & 54 & -13l_i\\ 22l_i & 4l_i^2 & 13l_i & -3L_i^2\\ 54 & 13l_i & 156 & -22l_i\\ -13l_i & -3l_i^2 & -22l_i & 4l_i^2\end{bmatrix}\begin{bmatrix}y_i''\\ \eta_i''\\ y_{i+1}''\\ \eta_{i+1}''\end{bmatrix}+\frac{E_i I_i}{l_i}\begin{bmatrix}12 & -6l_i & -12 & 6l_i\\ 6l_i & 4l_i^2 & -6l_i & 2l_i^2\\ -12 & -6l_i & 12 & -6l_i\\ 6l_i & -2l_i^2 & -6l_i & 4l_i^2\end{bmatrix}\begin{bmatrix}y_i\\ \eta_i\\ y_{i+1}\\ \eta_{i+1}\end{bmatrix}=\begin{bmatrix}f_{y_i}\\ M_{xi}\\ f_{y_{i+1}}\\ M_{xi+1}\end{bmatrix} \tag{3.3}$$

The set of equations 3.3 describes the lead-lag motion, the set of equations that describe the up-down bending motion can be obtained similarly.

These sets of equations 3.1, 3.2 and 3.3 are just for one element but can be extended to describe all elements that describe a beam or rotor blade. The result is a set of first order ordinary differential equations, for each motion, that can be written as:

$$Mu''+Ku=F \tag{3.4}$$

with M being the inertia matrix (masses and mass moments of inertia's), K the stiffness matrix (torsional, longitudinal and bending), F the external forces and moments vector and u the displacement vector (angles and displacements).

The accuracy of the Ritz-Hamilton model depends on the number of elements that are used [8–13]. Inman shows that for this model the error of the calculated natural frequency of the first mode shape of a uniform beam is under 11% when using only one element [10]. Increasing the number of elements to three reduces this error to less then 1% [10]. The errors for the natural frequencies of the higher mode shapes and for non uniform beams are less accurate, but improve similar with increasing number of elements.

3.1.3. IMPLEMENTATION OF THE STRUCTURAL MODEL

The structural model consists of two functions: Fo_struc and $KMmatrix$ and requires two input files: the $input\_struc.csv$ file containing structural data and the general input file $inputfile.py$.

The Fo_struc function imports the data from the structural input file and creates a new structural data file with structural data for each element. This allows the structural input file to have a different step size number than uses for the aeroelastic model. This means it is not necessary to change the structural input file if the step size is changed. It is also not necessary to perform measurements at every points. A copy of this new structural data file is saved to the results folder (defined in $inputfile.py$).

The $KMmatrix$ function uses this new structural data to calculate the K_{single} and M_{single} matrices for each separate element. These single matrices are then combined to create a K and a M matrix for a complete rotor blade. The result is four K-matrices and four m-matrices, one set for each motion: the torsion motion, the elongation motion, the flap bending motion and the lead-lag bending motion.

3.2. AERODYNAMIC MODEL

An aerodynamic model describes the aerodynamic forces acting on a wing or rotor blade. In the first subsection of this section the selection of a structural model is explained. In the second subsection the theory of the method of choice is described. In the final subsection the implementation of the theory into code is described.

3.2.1. MODEL SELECTION

Several aerodynamic models, like the blade element method, the lifting line method, the lifting line quasi steady method, the Navier-Stokes equations and the Euler equations were considered. A lifting line, quasi steady model delivers more accurate results compared to the blade element method and the lifting line model, while it is less complex to implement compared to the Navier-Stokes and Euler equations based methods. It captures 3D effects and includes effects due to the movement of the flow and the rotor blades. These movements not only reduce the effective angle of attack, they also have a stabilizing effect (aerodynamic damping) on the movement of the blade [1, 16]. Therefore the lifting line, quasi steady model was found to be most suitable to use for this project.

3.2.2. MODEL DESCRIPTION

In a lifting line, quasi steady model lift, drag and aerodynamic moment for a rotor blade are given by [1, 17–20]:

$$\begin{aligned} L &= \tfrac{1}{2}\rho S \int_0^R C_l \Omega^2 r^2 dr \\ D &= \tfrac{1}{2}\rho S \int_0^R C_D \Omega^2 r^2 dr \\ M_a &= \tfrac{1}{2}\rho S \int_0^R C_m \Omega^2 r^2 dr \end{aligned} \tag{3.5}$$

In the equations for lift and aerodynamic moment the lift and moment coefficients for airfoils (2D) are used. For aircraft wings these coefficient are usually corrected into lift and moment coefficients for finite wings (3D). However for rotorcraft applications two dimensional data is used almost exclusively [1]. So there is no correction for 3D effects on the lift curve, however an inflow model is used to correct for the reduced effective angle of attack. The drag coefficient used is corrected for 3D effects.

LIFT AND MOMENT COEFFICIENT

The lift and moment coefficients are corrected for a reduced effective angle of attack. This reduced effective angle of attack is given by [1, 16]:

$$\alpha = \theta - \theta_1 - \theta_2 \tag{3.6}$$

In which θ is the incidence angle of the blade with respect to the rotor plane, θ_1 is the angle caused by movement of the helicopter v_{h_z} and the up-down movement of the blade v_{r_z} with respect to the helicopter. The angle θ_2 is caused by the induced velocity of the air flow v_i around the blade.

Because v_{h_z}, v_{r_z} and v_i are small compared to Ωr equation 3.6 can be approximated by [1, 19]:

$$\alpha = \theta - \frac{v_{h_z} + v_{r_z} + v_i}{\Omega r} \tag{3.7}$$

The value for the induced velocity v_i can be calculated using the actuator disk theory [1, 19].

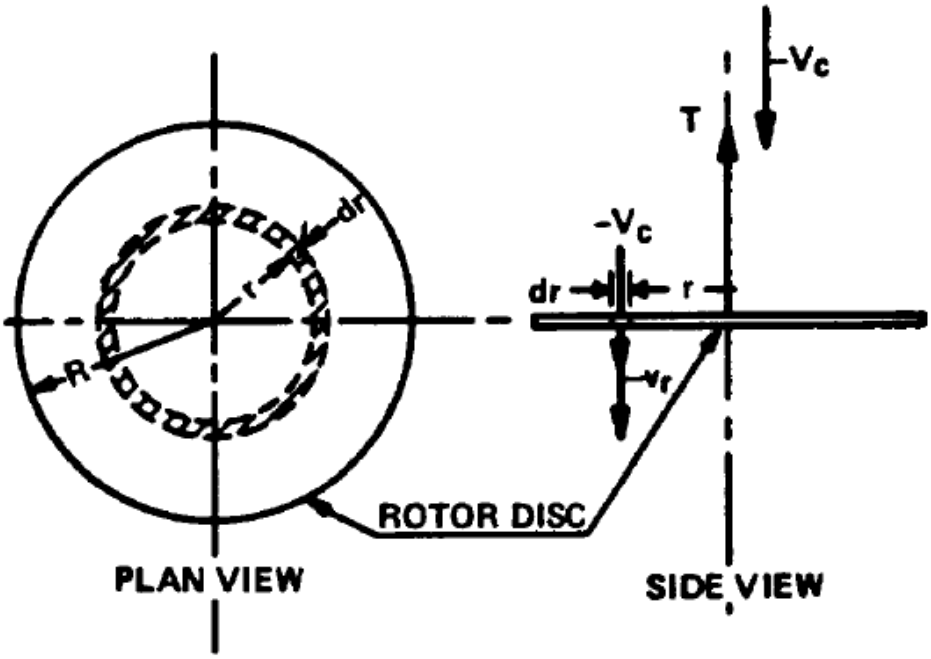


Figure 3.2: Elementary ring of the rotor disc [1]

From momentum theory the thrust generated by a ring with an area of $2\pi r dr$ as displayed in figure 3.2 can be derived:

$$dT = 2\pi\rho(v_{h_z} + v_{r_z} + v_i)v_i r dr \tag{3.8}$$

From the lifting line theory the thrust generated by a wing with an area of cdr is given by:

$$dT = \tfrac{1}{2}\rho C_L(\Omega r)^2 c dr \tag{3.9}$$

If the movement of the helicopter and the up and down movement of the blade with respect to the helicopter are neglected equation 3.8 and 3.9 can than be combined and written as:

$$v_i = \sqrt{\frac{C_L\Omega^2 cr}{4\pi}} \tag{3.10}$$

DRAG COEFFICIENT

Equations 3.5 use a drag coefficient for a finite wing or rotor blade C_D. This C_D is a function of the drag coefficient C_d for the airfoil (2D), the lift coefficient C_L, the Oswald factor e and the aspect ratio AR [17]:

$$C_D = C_d + \frac{C_l^2}{\pi e AR} \tag{3.11}$$

3.2.3. IMPLEMENTATION OF THE AERODYNAMIC MODEL

The aerodynamic model consists of five functions: $swashplate$, $aero\_fo$, vi_f, vi_f2 and $aero$. The $swashplate$ function provides the collective angles and rotational speed. The $aero\_fo$ function creates interpolation functions for chord length, twist angle and the lift, drag and moment coefficient. The vi_f and vi_f2 functions are two inflow models. These inflow models calculate the induced velocity of the flow. The $aero$ function combines the results of the other functions in the aerodynamic model and calculates the aerodynamic performance.

SWASHPLATE FUNCTION

The $swashplate$ function returns the angle of the root, the rotational speed, the collective angle, cyclic amplitude, the azimuth angle and the forward velocity of the rotorcraft. It uses a time vector as input, a swashplate input file and the general $inputfile.py$. The location and name of the swashplate input file is defined in $inputfile.py$. An example of a swashplate input file is given in appendix C.

The $swashplate$ function receives a time vector by the instance calling the function. The collective angle, the swashplate amplitude, the rotational speed and forward speed of the rotor craft are defined in the swashplate input file. The $swashplate$ function reads the swashplate input file and creates linear interpolations for the rotational speed, the collective angel, the swashplate amplitude and the forward speed of the rotor craft. The interpolation of the rotational speed is used to calculate the azimuth angle. Next the $swashplate$ function calculates the angle of the root section as a function of the time vector, using equation 3.12.

$$\alpha_{root} = \alpha_{coll}(t) + \alpha_{ampl}(t) \cdot \sin(\Psi(t)) \tag{3.12}$$

In this equation α_{root} is the angle of the root. The twist distribution is not taken into account in by function. This is done in the $aero$ function.

INTERPOLATION FUNCTION

The function $aero\_fo$ function creates the interpolation functions for chord length, twist angle and the lift, drag and moment coefficient. It uses a set of airfoil data files, an aerodynamic input file and the general input file. The set of airfoil data files contains the aerodynamic coefficients as function of r and α. The aerodynamic input file contains twist and chord data. Examples of the aerodynamic input file and a file out of the set of airfoil data files are given in appendix C. The process of creating a set of airfoil data files is describes in chapter 4.

The chord and twist are linear interpolated as a function of r. A surface is fitted through the aerodynamic coefficients as a function of r and α. These interpolations and fitted surfaces functions allow the aeroelastic model to use values for the twist angle and the aerodynamic coefficients as a function of values of r and α in between the values described in the aerodynamic input file and the set of airfoil data files. Figure 3.3 shows a 3D representation of the C_L, r and α data from the airfoil data set. Figure 3.4 shows the fitted surface.

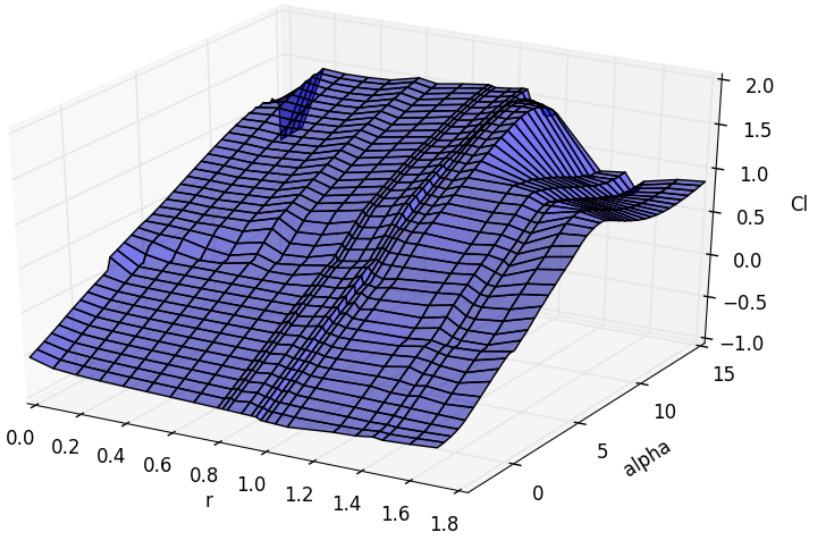


Figure 3.3: Cl as a function of r and α from Xfoil data files

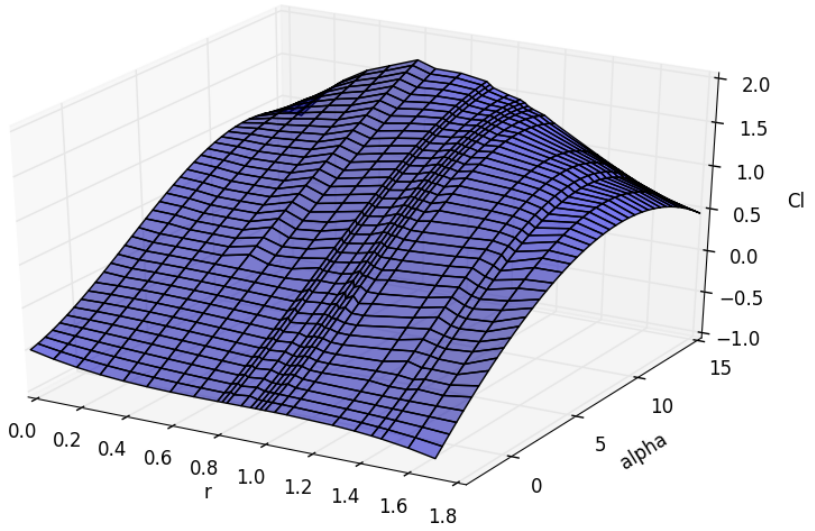


Figure 3.4: Cl as a function of r and α, from interpolation function

INFLOW MODEL: CONSTANT FRACTION

The third function, vi_f, calculates the induced velocity of the flow. The induced velocity is given by equation 3.10, however solving equation 3.10 is not straght forward because C_L depends on v_i.

A estimation of the induced velocity v_i^* (the * designation is used to distinguish estimates from actual values) can be made by repeating the following steps:

1. Use $v_i^* = 0$ in equation 3.7. Because v_i^* is lower than v_i α^* will be higher than α.
2. Use the interpolation function for C_L to find C_L^*. Because α^* is higher than α C_L^* will be higher than C_L
3. Use this value of C_L^* in equation 3.10. Because C_L^* is higher than C_L v_i^* will be higher than v_i.
4. Use this value of v_i^* in equation 3.7. Because v_i^* is higher than v_i α^* will be lower than α
5. Use the interpolation function for C_L to find C_L^*. Because α^* is lower than α C_L^* will be lower than C_L

6. Use this value of C_L^* in equation 3.10. Because C_L^* is lower than C_L v_i^* will be lower than v_i.
7. Use this value of v_i^* in equation 3.7. Because v_i^* is lower than v_i α^* will be higher than α
8. Repeat the process from step 2

If this process is repeated alternating over predictions and under predictions for v_i^*, α^* and C_L^* are produced. These process was performed for many angles of incidence and at many distances from rotational centre. The results of these calculations showed that these over predictions and under predictions converge into a single value, for each variable. Not only do these predictions converge into a single value, this converged value can be approximated as constant fraction of the start variable (corresponding to $v_i = 0$). An sample of the results of these calculations is shown in table 3.1.

r	$C_{L_{vi=0}}$	C_L	$C_L/C_{L_{vi=0}}$
0.4250	1.208	0.529	43.8%
0.6375	1.182	0.549	46.4%
0.8500	1.143	0.523	45.7%
1.0625	1.076	0.493	45.8%
1.2750	0.991	0.473	47.7%
1.4875	0.904	0.434	48.0%
1.7000	0.835	0.425	50.9%
0.4250	1.209	0.530	43.8%
0.6375	1.183	0.549	46.4%
0.8500	1.144	0.523	45.7%
1.0625	1.081	0.497	46.0%
1.2750	0.995	0.476	47.8%
1.4875	0.897	0.429	47.8%

Table 3.1: Induced velocity for 7° angle incidence.

INFLOW MODEL: STEPNIEWSKI

Stepniewski's inflow model [1] is implemented for comparison. Stepniewski proposes a inflow model based on the Cl_α slope. For a linear twisted blade with a constant chord his inflow model becomes:

$$v_i = v_t \left[-\frac{Cl_\alpha \sigma}{16} + \sqrt{\left(\frac{Cl_\alpha \sigma}{16}\right)^2 + \frac{Cl_\alpha \sigma}{8} \bar{r}(\theta_0 - \theta_{tot}\bar{r})} \right] \tag{3.13}$$

Which can be written as:

$$v_i = v_t \left[-\frac{Cl_\alpha \sigma}{16} + \sqrt{\left(\frac{Cl_\alpha \sigma}{16}\right)^2 + \frac{Cl_\alpha \sigma}{8} \frac{r}{r_{max}} \theta(r)} \right] \tag{3.14}$$

Equations 3.13 and 3.14 predict that the flow velocity is zero for an angle for zero effective angle of attack. This is only correct for symmetrical airfoils. Therefore equation 3.14 is adjusted:

$$v_i = v_t \left[-\frac{Cl_\alpha \sigma}{16} + \sqrt{\left(\frac{Cl_\alpha \sigma}{16}\right)^2 + \frac{Cl_\alpha \sigma}{8} \frac{r}{r_{max}} (\theta(r) - \theta_{L=0})} \right] \tag{3.15}$$

COMPARISON OF THE INFLOW MODELS

The inflow model is selected by lines in the *aero* function in the Aero_Model.py file. In section 3.2.3 instructions are given on how to select the either of the two inflow models. A comparison of the results of both inflow models is shown in table 3.2.

α	Constant fraction model	Stepniewski's model	Stepniewski's/Constant fraction
-0.4	3.263	2.590	79.4%
0.6	4.654	3.934	84.5%
1.7	5.496	4.620	84.1%
2.7	5.866	4.826	82.3%
3.8	5.778	4.624	80.0%
4.1	9.010	8.117	90.1%
4.8	9.306	8.550	91.9%
5.5	9.767	9.167	93.9%
6.8	9.939	9.553	96.1%
7.7	9.658	9.227	95.5%
8.7	9.120	8.558	93.8%
8.7	9.072	8.513	93.8%
9.8	8.353	7.634	91.4%
11.1	7.240	6.426	88.8%

Table 3.2: Induced velocity predicted by different inflow models.

Table 3.2 shows that the results of the models does not match exactly, but are roughly in agreement. Stepniewski's model predicts a slightly lower flow velocity. As a result the effective angle of attack is less affected, resulting in slightly more lift, drag and power required, when using this inflow model.

AERODYNAMIC PERFORMANCE FUNCTION

The $aero$ function uses the interpolation functions obtained by $aero\_fo$ and data in the input file to calculate aerodynamic performance. It receives the interpolation functions and the data from the Commander (command prompt or script). The $aero$ function calls either of the two inflow model functions to obtain the induced velocity of the flow.

At the start of the $aero$ function the r vector is created. Next all variables that will be used for calculations are declared. Next the neutral point data is read from the structural file.

Next the rotational speed for each section is calculated. Note that due to deformations the rotational speed does not need to be the same for each element. Using the rotational speed and the radius, the speed of the airflow is calculated for each element. The aspect ratio for the whole blade is calculated.

Than a loop is started, in which values are calculated for sections with an increasing r. The angle of attack, with $v_i = 0$ is calculated for each section. The $C_{l\alpha}$ for that section is calculated. The first is needed for both the constant fraction inflow model" and Stepniewski's inflow model, the second only for the Stepniewski's inflow model.

Than either of the two inflow models is used based on which of the following lines are commented:

```
267         # Choose an inlfow model:
268         # Constant fraction inflow model:
269         vi = vi_f(tck_cl, r[i], alpha[i], c[i], Omega[i], ge_f, vi_coef)
270         # Stepniewski's inflow model:
271         #vi = vi_f2(r[i], rmax, Omega[i], ge_f, alpha[i], c[i], nr_b, Cla)
272         #vi2 =vi_f2(r[i], rmax, Omega[i], ge_f, alpha[i], c[i], nr_b, Cla)
273         #print alpha[i], vi, vi2, vi2/vi
```

Line 269 defines the constant fraction inflow model to be used. Line 271 defines Stepniewski's inflow model to be used (but is commented in the example above, so it won't be used). Line 272 also uses Stepniewski's inflow model, but writes results to another variable, which is useful for comparing the two inflow models. Line 273 prints this comparison, which was used to create table 3.2.

When v_i is returned from the inflow model function the angle of attack can be calculated using equation 3.6. Using the angle of attack and the interpolation functions defined by $aero\_fo$ the aerodynamic coefficients can be determined. A $3D$ correction using equations 3.11 is performed. Using the aerodynamic coefficients, the velocity and the air properties defined by the general input file, the lift, drag, aerodynamic

moment and power required for that element can be determined.

The calculated values are written to the F_z and F_y vectors which were declared at the beginning of the function.The torsion is calculated by adding the moment due to lift to aerodynamic moment and written to the F_T vector.

3.3. OTHER FORCES

In the previous sections elasticity and aerodynamic forces are described. Other significant forces that act on rotor blades are centripetal forces and forces due to conservation of momentum. In the first subsection the theory is described. In the final subsection the implementation of the theory into code is described.

3.3.1. DESCRIPTION

The theory about centripetal force and conservation of angular momentum is described here.

CENTRIPETAL FORCE

If an object rotates around a point the centripetal force accelerates the mass towards the centre of rotation. Without centripetal force the object would not rotate arround the point but just drift away. Centripetal force is not only a very large force, it is also a stabilizing contribution. This stabilizing effect reduces the bending moment around the root of the rotor blade considerably [1, 19].

The centripetal force required to keep an object with mass m and tangential speed v at a curved path with radius r is given by:

$$F = \frac{mv^2}{r} \tag{3.16}$$

For a path with constant radius, like that of an element of a rotorblade, the speed can be written as:

$$v = \Omega r \tag{3.17}$$

A drawing of an element of a blade is shown in figure 3.5. The element has a mass m_n and its centre of gravity is at a distance r from the rotational centre, x in the rotational plane and z perpendicular to the rotational plane. The values of x and z are a function of r, the angle at which the blade is fixed and the deformation of the blade.

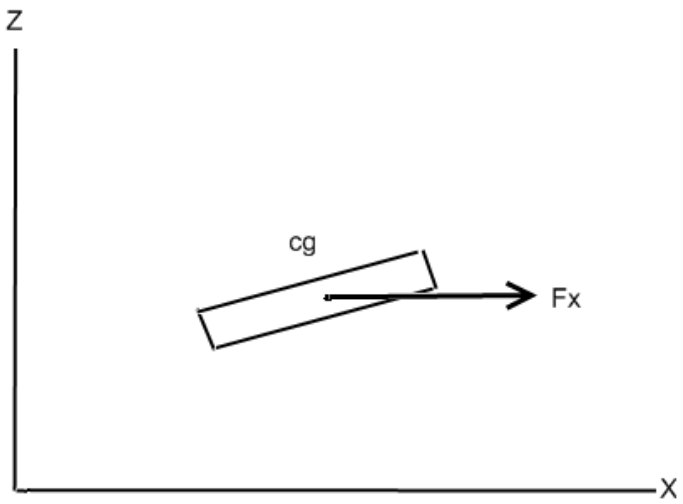


Figure 3.5: Centripetal force on an element.

The centripetal force on this element is given by:

$$F_x = m_n \Omega^2 x_n \tag{3.18}$$

This force creats a moment around the root of:

$$M_y = z_{cg} F_x \tag{3.19}$$

Where z_{cg} can be expressed as the midpoint between the two nodes:

$$z_{cg} = 0.5(z_n + z_{n+1}) \tag{3.20}$$

When combined, this can be written as:

$$\begin{bmatrix} f_{z_n} \\ M_{y_n} \\ f_{z_{n+1}} \\ M_{y_{n+1}} \end{bmatrix} = F_x \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \end{bmatrix} \begin{bmatrix} z_n \\ \beta_n \\ z_{n+1} \\ \beta_{n+1} \end{bmatrix} \tag{3.21}$$

This set of equations can be written as [12]:

$$F_{cp} = K_{cp_z} u \tag{3.22}$$

The force in x direction is given by a F_{cp_x}-matrix. The values in this matrix can be directly calculated from equation 3.18. The stabilizing moment around the z axis can be derived using the same method.

CONSERVATION OF ANGULAR MOMENTUM

A beam is displayed in figure 3.6. This left picture is the initial state and displays a horizontal bar rotating around the z axis. In the second picture the bar is rotated α_y around the y axis.

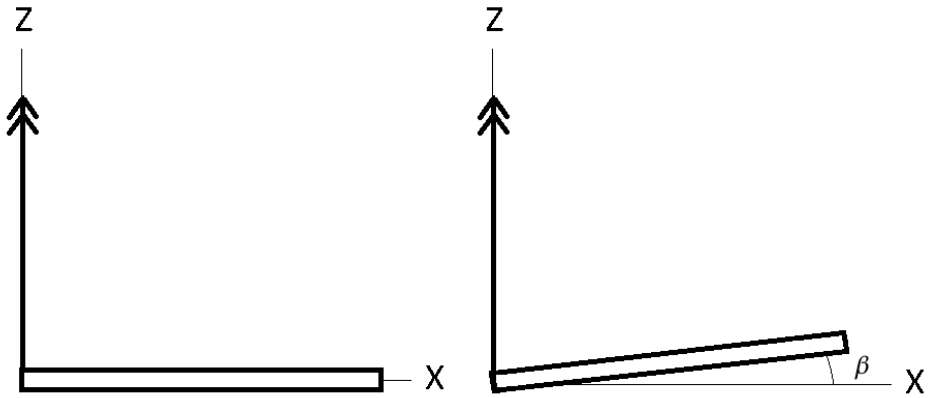


Figure 3.6: Coriolis effect.

For the left picture the angular momentum in z-direction equals:

$$M_z = \frac{1}{2} I_{zz} {\Omega_z}^2 \tag{3.23}$$

In the right picture the angular momentum in z-direction equals:

$$M_z = \frac{1}{2} I_{zz} {\Omega_z}^2 \cos\beta \tag{3.24}$$

If there is no external torque applied the conservation of momentum theorem states that expression 3.23 and 3.24 should be equal. So with increasing α the rotational speed Ω_z has to increase in order for the momentum M_z to remain constant. A positive force (in y direction) is working on the bar, resulting in a positive

moment in z direction, causing Ω_z to increase.

For small angles the force on an element of the blade will be:

$$M_z = \frac{1}{2} I_{zz} {\Omega_z}^2 \beta' \tag{3.25}$$

The force on an element of the blade with mass m_n at distance x_n will be:

$$F_y = \frac{1}{2} m_n {x_n}^2 {\Omega_{z_n}}^2 \beta'_n \tag{3.26}$$

3.3.2. IMPLEMENTATION OF THE FORCES MODELING

This subsection describes the implementation of the forces described by the previous subsection. These force functions all use the structural input file, the general inputfile and the function Fo_struc described in section 3.1.3.

CENTRIPETAL FORCE

At the start of the centripetal force function the number of elements is calculated and variables are declared. The structural data, obtained by the Fo_struc function is assigned to variables. Next the positions of the centres of gravity and the corresponding rotational speed is calculated. The centripetal force per element is calculated using equation 3.21. This is assigned to ${F_{cp}}_x$. The torsion moment due to the centripetal force is calculated and assigned to ${F_{cp}}_T$. Finally ${K_{cp}}_z$ and ${K_{cp}}_y$ are calculated as described in section 3.3.1.

CONSERVATION OF ANGULAR MOMENTUM

There are two momentum functions: Fmomentum and Fmomentum2. Fmomentum only returns zero force vectors. If this function is used the contribution of conservation of angular momentum is discarded. This function can be used to reduce calculation time. Fmomentum2 does include contributions of conservation of angular momentum and is described here.

First the number of elements and nodes is calculated. Next empty variables x, $E_x A_x$, ρA_x, y_{np}, y_{cg}, I_{zz} and β' are created. The content from the structural data file are assigned to these variables. Next the rotational speed is calculated and the force matrices are declared. Forces due to momentum are calculated using equation 3.26 and assigned to the force matrices.

GRAVITY

At the start of the gravity function the number of elements is derived. With this value, the variables can be declared next. Similar as with the centripetal force the structural data from the Fo_struc function is assigned to these variables. The position of the nodes is calculated. The force and torsion vectors F_T, F_x, F_y and F_z are declared. The gravity over each element is calculated and assigned to the node at the root side of the element, together with a moment of $F_z l_e$. The torsion due to gravity is calculated using y_{cg} and y_{np} data.

3.4. INTEGRATION OF MODELS AND CALCULATING RESULTS

The previous sections described the structural, aerodynamic and force models. These parts of the aeroelastic model need to be combined and solved together to obtain the results described in chapter 1. In the first subsection the theory is described. In the final subsection the implementation of the theory into code is described.

3.4.1. DESCRIPTION OF INTEGRATION OF MODELS

The structural model was given by equation 3.1.2 and given by:

$$Mu'' + Ku = F(t) \tag{3.27}$$

The centripetal force was given by equation 3.22

$$F_{cp} = {K_{cp}}_z u \tag{3.28}$$

The other forces, including aerodynamic, can be written as:

$$F = F(t) \tag{3.29}$$

By combining equations 3.27, 3.28 and 3.29 the aeroelastic model is given by:

$$u'' = Au + F(t) \tag{3.30}$$

with:

$$A = M^{-1}(K - K_{cp}) \tag{3.31}$$

From this point the process of obtaining the results described in chapter 1 can be split into two parts. The first part is obtaining the natural frequencies of the system described by equation 3.4.1. This can be done by finding the Eigenvalues of the A matrix. The second part is obtaining the response of the system. The response can be obtained by integrating equation 3.4.1 twice. The results of this method describe the response of the system on a given input and include the forces on the rotor blade and the deformations of the rotor blade.

NATURAL FREQUENCIES AND STABILITY OF THE SYSTEM

The natural frequency or resonance frequency is the frequency at which a part or an assembly vibrates freely (without external forces). If these natural frequencies match the frequency of external forces resonance occurs. This can cause a product to fail even though it is loaded below its static strength. An famous example of this is the Tacoma Narrows bridge collapse in 1940. It is therefore very important to create an offset between the natural frequency and the loading frequencies.

To find the frequency at which a linear system vibrates freely, the forces that do not depend linearly on the state of the system $F(t)$ are neglected. Equation 3.4.1 without the external forces as a function of t reduces to:

$$u'' = Au \tag{3.32}$$

The natural frequencies of this system described by equation 3.4.1 are given by the Eigenvalues of matrix A. This A matrix depends on the rotational speed, because K_{cp} depends on the rotational speed. Therefore the natural frequencies can be plotted against the rotational speed. This is called a Campbell diagram [21].

RESPONSE OF THE SYSTEM

The response u of the system described by equation 3.4.1 was obtained by integrating equation 3.4.1 twice. A numerical integration method was chosen because of it can be used very well to solve a wide range of complex problems by a computer.

Several numerical integration methods were considered, like the Euler's method, the mid-point method and the class of Runge-Kutta methods. The order of error and the (computational) complexity of these methods were investigated. The Euler's method is the most simple method. It requires the least amount of calculations, but it has an error of order h [22]. The mid-point method requires a few more calculations, but the order of error is improved to h^2 [22]. A fourth order Runge-Kutta method requires even more calculations but it's error is of order h^4 [22]. Although higher order integration method require more calculations per integration step, their higher order error makes it possible to user bigger step sizes, resulting in significant reduction in total calculation time [22]. Therefore the fourth order Runge-Kutta method was implemented. For comparison the mid-point method is also implemented.

The Runge-Kutta methods use the information of the slope at more than than one point to determine the u_{n+1}. The evaluation of u_{i+1} by a fourth order Runge-Kutta is given by [22]:

$$u_{i+1} = u_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3.33}$$

with:

$$\begin{aligned} k_1 &= f(t_i, u_i), \\ k_2 &= f(t_i + \tfrac{1}{2}h, u_i + \tfrac{h}{2}k_1), \\ k_3 &= f(t_i + \tfrac{1}{2}h, u_i + \tfrac{h}{2}k_2), \\ k_4 &= f(t_i + h, u_i + hk_3) \end{aligned} \tag{3.34}$$

The fourth order Runge-Kutta method is used by the $Integrate2$ function. This function is used by default, however the $Integrate$ function, which uses the mid-point method, can be used if desired.

The midpoint method uses the derivative between i and $i+1$ to calculate u_{i+1}. The evaluation of u_{i+1} by the mid-point method is given by [22]:

$$u_{i+1} = u_i + h \cdot u'_i + \frac{1}{2}h^2 \cdot u''_i \tag{3.35}$$

Equation 3.35 describes the response at $i+1$ as a function of u, u' and u'' at i.

3.4.2. IMPLEMENTATION OF INTEGRATION OF MODELS

In the first subsection the functions used for combining the previous described models are described. In the second part of this section the functions used during the calculation of results and the script to calculate these results are described.

COMBINE_F

The $combine\_F$ function transforms a force vector into another force vector with reduced dimensions. This function can be used for example to reduce the dimensions of the output of the aerodynamic model to match the dimensions of the structural model. This allows the aerodynamic model to be used with more elements than the structural model, without a dimension mismatch in equation 3.4.1.

EXPAND_F

The $combine\_F$ function reduces the force factor to accommodate the structural model, the $Expand\_F$ function does the opposite of that: Some force models, like the centripetal force model, require displacement of the structural model as input. If the structural model that provides the displacements uses less elements than the force models this will be a problem. That's where the $Expand\_F$ function comes in. It rewrites a coordinate vector (like displacements) to a coordinate vector with increased, so it can be used by the force models. It does this by interpolation between the coordinates of the input vector.

BC

The BC function is actually a boundary conditions input file and is used in the integration process. It requires the number of nodes as input and returns the deformation vectors at $t = 0$: x_0, x'_0, y_0, y'_0, z_0, z'_0, T_0 and T'_0. All deformations at $t = 0$ are set to zero by default, however another boundary condition may be defined in this function if needed.

INTEGRATE

The integrate function calculates u_{i+1}, u'_{i+1} and u''_i, using a mid-point method. The integration process consists of three steps:

1. In the first step u''_i is calcuated using u'_i, u_i, the force vector F_i and equation 3.4.1.
2. In the second step u_{i+1} is calculated using u''_i, u'_i, u_i and equation 3.35. This part is the important part for the response calculations.
3. In the third step u'_{i+1} is calculated. To accomplish this, first $u_{i+1/2}$ is calculated as the mean of u_i and u_{i+1}. With $u_{i+1/2}$ and equation 3.4.1 $u''_{i+1/2}$ is obtained. Now u'_{i+1} can be calculated using the mid-point method.

INTEGRATE2

The integrate function calculates u_{i+1}, u'_{i+1} and u''_i, using a fourth order Runge-Kutta method. The integration process consists of five steps:

1. In the first step $k1$ is calculated for u_{i+1} and u'_{i+1} using equation 3.34.
2. In the first step $k2$ is calculated for u_{i+1} and u'_{i+1} using equation 3.34.
3. In the first step $k3$ is calculated for u_{i+1} and u'_{i+1} using equation 3.34.
4. In the first step $k4$ is calculated for u_{i+1} and u'_{i+1} using equation 3.34.
5. In the fifth step $k1$, $k2$, $k3$, $k4$ and $k5$ are combined and equation 3.33 used to calculate u_{i+1} and u'_{i+1}.

CAMPBELL SCRIPT

The purpose of this script is to calculate and plot the natural frequencies as a function of the rotational speed. This plot is called a Campbell diagram. The natural frequencies of the system are given by the eigenvalues of the A-matrix in equation 3.4.1. These values depend on the rotational speed due to centripetal force as stated in equation 3.4.1.

At the begin of the Campbell script a timer is started. The input file, Python modules and scripts are imported. Scripts the Campbell scripts uses are: Structural_Model.py, Forces_Model.py and BC_f.py. A folder to write results to is created, the used input file is copied there and a log file is created

Next a structural data file is created using the FO_struc function, after which $K_{structural}$ and M matrices are created using the $KMmatrix$ function. The number of steps, defined by the input file, is calculated and the files to store the natural frequencies are defined. All deformations are defined and set to zero using the BC_f function.

The natural frequencies are calculated in a loop that increases the rotational speed each step. The start point, end point and number of steps are defined in the input file. The loop calculates the rotational speed rpm defined by the number of steps and the values from the input file. Next the centripetal force function is used to calculate $K_{y_{cp}}$ and $K_{z_{cp}}$. The K_{struc} matrices are added to the K_{cp} matrices after which the A matrices are calculated, for each motion. The Eigen frequencies of the A matrices are calculated, sorted and assigned to variables.

Now the natural frequencies are know they are plotted against the rotational speed in a Campbell diagram. The log file is closed and saved to results dir.

RESPONSE SCRIPT

The response of a system u_i can be obtained by integrating equation equation 3.4.1. However due to a very small step size in time required for stability the size of u will be very large. This caused a lot of memory use, errors and eventually caused the program to crash. Therefore the data to be plot and the data used for integration is decoupled.

At the begin of the Response script a timer is started. The input file, Python modules and scripts are imported. Scripts the Response scripts uses are: Structural_Model.py, Integrate_f.py, Aero_Model.py, Forces_Model.py and BC_f.py. Next the program checks whether correct parameters are defined in the input file. For example: it is not possible to have more points to plotted ($nplot$) than integration steps (t_{end}/h_t).

A folder to write results to is created, a copy of the used input file is saved and a log file is created. Next a time vector is created. The $swashplate$ function is used to obtain the angels of the root section. A copy of the swashplate-input file, defined in the input file is saved to the result directory. A new structural data file is created using the FO_struc function, after which K_{struc} and M matrices are created using the $KMmatrix$ function.

The variables that will be plotted and the variables that are needed for calculations are declared. All initial deformations are set to zero using the BC.py function. The aerodynamic interpolation functions are created using the $aero\_fo$ function. A copy of the aerodynamic input file is saved in the result folder.

A loop is started in which the integration is calculated:
Lift, drag, aerodynamic moment, moment due to lift and power of each element is calculated using the $aero$ function. Centripetal force function is used to calculate $K_{y_{cp}}$, K_{zcp}, F_{Tcp} and F_{xcp}. The $Fmomentum$ and $Fgravity$ functions are used to calculate $F_{Tm}, F_{xm}, F_{y_m}, F_{zm}$ and $F_{Tg}, F_{xg}, F_{y_g}, F_{zg}$. Forces are added up, K-matrices (structural and centripetal force) are combined. The A-matrices and natural frequencies are calculated.

Next all four sets of equations (torsion, longitudinal, flap and lead-lag) are integrated using the $Integrate$ (mid-point method) function or the $Integrate2$ (fourth order Runge-Kutta method) function. The integration are performed in a loop in which u_{calc} is evaluated. This u_{calc} consists off u_i and u_{i+1}. Using equation 3.34 or equation 3.35 u_{i+1} is calculated. At the end of the loop $u_{n,i+1}$ is copied to $u_{n,i}$ for the next run. So only the current and previous values of u are stored. A counter keeps track of how many times the loop has run. If a certain value (based on values in the input file) is reached it writes u_i to u_{plot}. This data is used for plotting. This decoupling keeps u_{calc} small and u_{plot} smaller (compared to plotting every calculated value of u).
Inside this integration loop is an another loop. This second loop has a counter and if a specified value is reached it writes the integrated values to the plot files. These specified values determine the ratio between the number of points calculated and the number of points plotted. These specified values are specified and can be adjusted in the general input file.

At the end of the first loop the calculated values for T_{n1}, x_{n1}, y_{n1} and z_{n1} are copied to T_{n0}, x_{n0}, y_{n0} and z_{n0} for the next iteration.

After the integration loop the figures are plotted, saved and closed. The force files are saved to result folder. They can be used in ANSYS or a similar application for additional calculations if required. The log file is saved to the result folder.

RESPONSE2 SCRIPT

The purpose of the $Response2.py$ script is similar to that of the response.py script. It uses the same functions and writes the same files. The difference is that this script decouples the step size used for the structural model and the step size used for these aerodynamic model or force models.

The $Response2.py$ script has big advantage: it is possible to keep the number of elements for the structural model low. As a result the time step size can be bigger. At the same time more elements can be used for the aerodynamic model to ensure a realistic and gradual change in air flow velocity. The bigger time step will reduce the time needed for calculation significantly, while more elements used for the aerodynamic model means it remains its accuracy.

However there is a disadvantage: It is not possible to combine a K and M matrix and a F vector with non-matching dimensions directly. Furthermore some force functions require a displacement or velocity vector as input. It is a problem if the dimensions of the structural model and the force models do not match. To overcome this the $combine\_F$ and $expand\_F$ functions are used.

The goal of this script is to save calculation time while remaining accuracy. The $Response2.py$ script requires the use of the extra functions, which means it requires extra calculations compared to the $Response.py$ script when the same structural element size is used. This means it will take more time to execute. The $Response2$ script is only beneficial when a small force element size is combined with a larger structural.

4

TEST CASE

One of KVE's rotor blade project is the development of after market rotor blades for the Schiebel Camcopter S-100 operated by the Abu Dhabi Autonomous Systems Investments Company (ADASI) (Abu Dhabi, United Arab Emirates). KVE provided several of these rotor blades to perform the tests described in section 4.4. The goal of these tests is to compare the results to the results of the aeroelastic model. The results of these tests can be found in chapter 5.

These rotor blades are $1.7m$ long, have a mass of approximately $2kg$ (see table 4.1). These rotor blades are made of foam (core), glass fibre (girders) and carbon fibre (skin). A picture of these rotor blades can be found on the cover of this thesis, in figure 4.2 and in figure 4.6. The Camcopter S-100 has an operating rotational speed of approximately 1200 rpm. It has two rotor blades and a maximum take off weight of 200kg.

The NLR determined the blades can be approximated by four parts and three airfoils. This is shown in figure 4.1. The section from the root until slightly more than half the rotor blade length consists of a single airfoil (RC412). In the next 10% of the rotor blade length this airfoil blends into a similar but thinner airfoil (RC410). In the third part this second airfoil blends into a third airfoil (RC310). The fourth part consists of only the third airfoil until the tip.

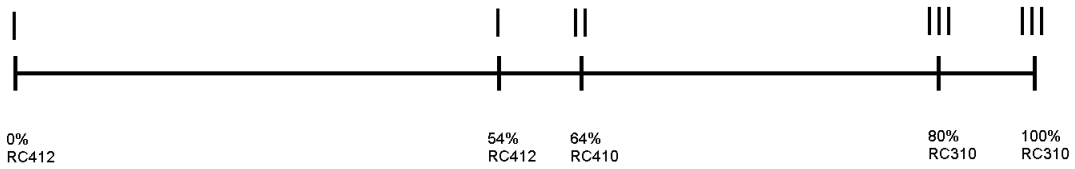


Figure 4.1: Profiles, distance in percent ofthe rotor blade length (not to scale)

In order to model these rotor blades, it was necessary to determine blade properties as input data for the aeroelastic model. The input data consists of aerodynamic sectional properties, structural sectional properties and inertial sectional properties.

4.1. AERODYNAMIC PROPERTIES

The aeroelastic model needs the following aerodynamic properties:

c, the chord length.
θ the incidence angle.
y_{ac}, the position of the aerodynamic centre.
C_L, the lift coefficient.
C_D, the drag coefficient.

C_M, the aerodynamic moment coefficient.

The chord length was not difficult to measure and is not explained here. The incidence angle is calculated using the root angle of incidence, the twist distribution and the torsion angle. The aerodynamic centre was considered to be constant and equal to the quarter chord length.

The values for C_L, C_D and C_M are described by a set of aerodynamic input files. An example of one input file of the set of aerodynamic input files is given in appendix C.5. For this test case Xfoil[23] was used. A python script (xfoil.py) is used to automate the generation of input files by using Xfoil.

The xfoil.py script contains a loop in which the value of r is increased stepwise for a defined range of r. It calculates values such as Mach number and Reynolds number as function of r, the chord length and the rotational speed. The script calls the Xfoil program for each value of r. The Xfoil programs generates an output file for a range of α, using the calculated Mach numbers and Reynolds numbers. The output files are saved in the "Results" folder, using the following name convention: [airfoil name]_[r]_[rpm].log. The range of r, the range of α, the airfoil coordinate file and other parameters can be changed in xfoil.py. A second script called xfoil_blend.py is similar as xfoil.py, but used for areas with blending airfoils. The locations of these airfoils (x coordinate) need to be defined in the script. It than calculates the blending-fraction and passes that to Xfoil.exe.

Figure 3.3 shows a 3D representation of the C_L, r and α data from the Xfoil data set.The first part ($r = 0$ to $r = 0.9$) is the single airfoil part. The data lines are evenly spaced. From $r = 0.9$ the data lines are closer to each other: that is the first interpolation part. The lines are closer because the step size was chosen smaller, because of the expected larger changes in C_L values. For the second interpolation Xfoil had troubles generating the data files. That is why there are less data lines in this area (between $r = 1.10$ and $r = 1.35$). The last part is a single airfoil area again and the data lines are more evenly spread.

4.2. STIFFNESS PROPERTIES

The aeroelastic model needs the following structural properties:

x The position of the node.
AE_x The stiffness in x-direction.
EI_{yy} Bending stiffness.
EI_{zz} Bending stiffness.
GI_p Torsional stiffness.
y_{nl} Position of the neutral line.
z_{nl} Position of the neutral line.

The bending and torsion stiffness of the blade was measured by a static deflection measurement as described by Jung et al [24] and by Anusonti-Inthra and Liu [25]. A measurement setup was designed for this purpose. Figure 4.2 shows one of the rotor blades in this measurement setup. In this figure the blade was fixed at the root (lower right corner). An aluminium bar was fixed at the rotor blade at a specified distance from the root. Weights (on the table) were applied at the ends of the aluminium bar, which caused the rotor blade to bend and twist. The displacements were measured at two positions using two callipers (above the aluminium bar).

By combining the applied loads and measured displacements an equivalent stiffness of the rotor blade can be obtained. It proofed to be very difficult to calculate the stiffness of the sections: small measurement errors caused big changes is stiffness, especially for small sections. Therefore these equivalent stiffness of the rotor blade were compared to an estimated stiffness based on structural data such as changes in circumference, changes in main girder thickness, changes in skin layup and others. This estimated stiffness distribution is than altered to match the measurement results. Figures 4.3, 4.4 and 4.5 show the measured and adjusted estimated stiffness distribution. These figures show it is possible to obtain a stiffness distribution that both matched expectations based on structural knowledge and the measurement results.

Figure 4.2: Measurement setup

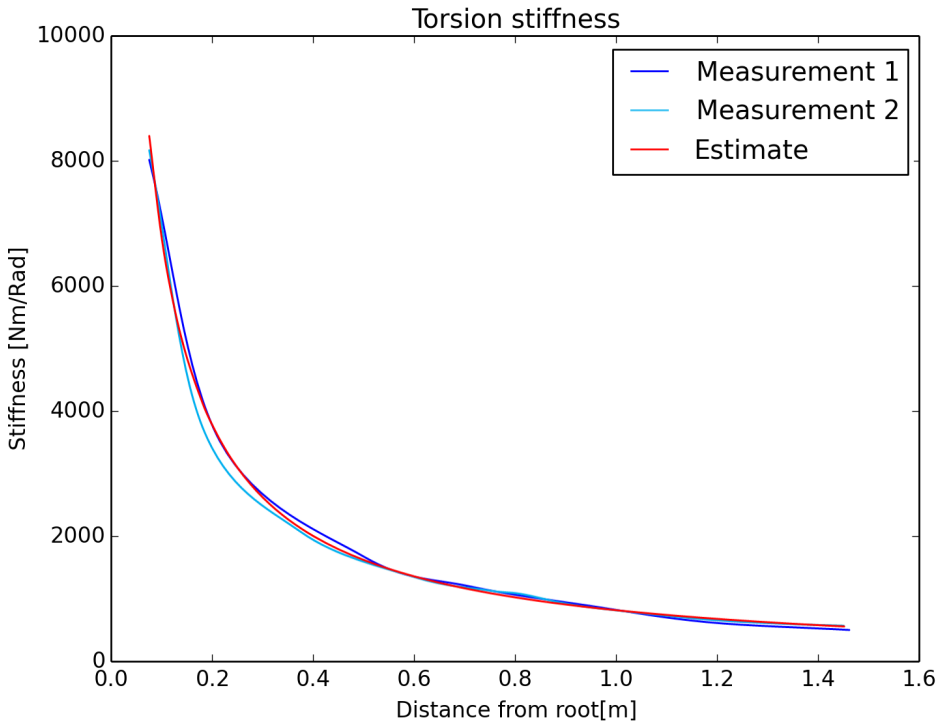


Figure 4.3: Torsional stiffness

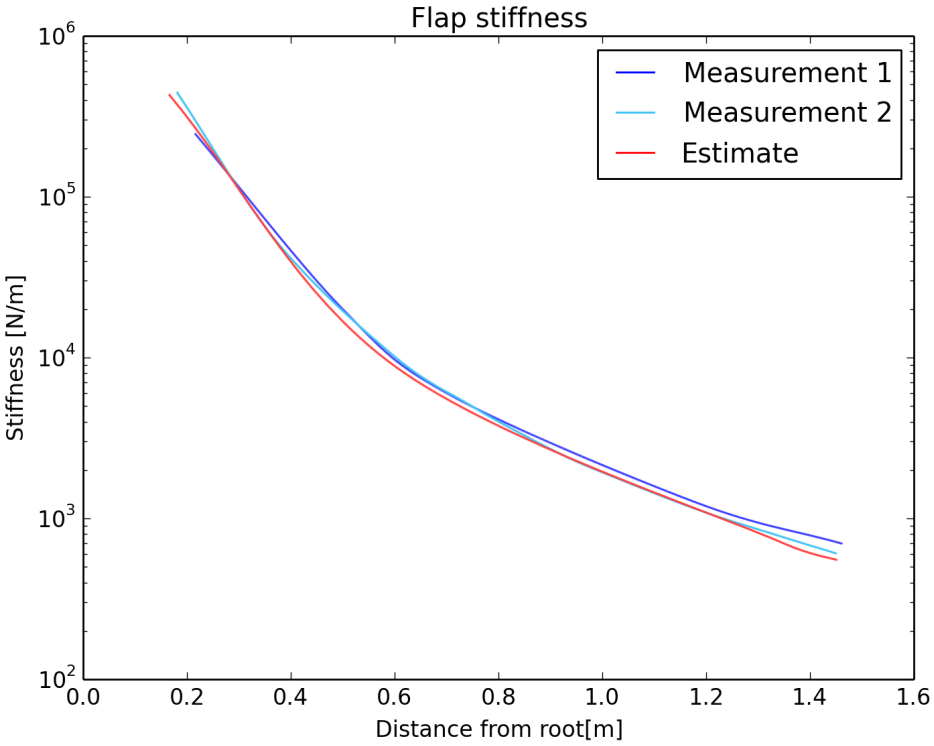


Figure 4.4: Flap bending stiffness

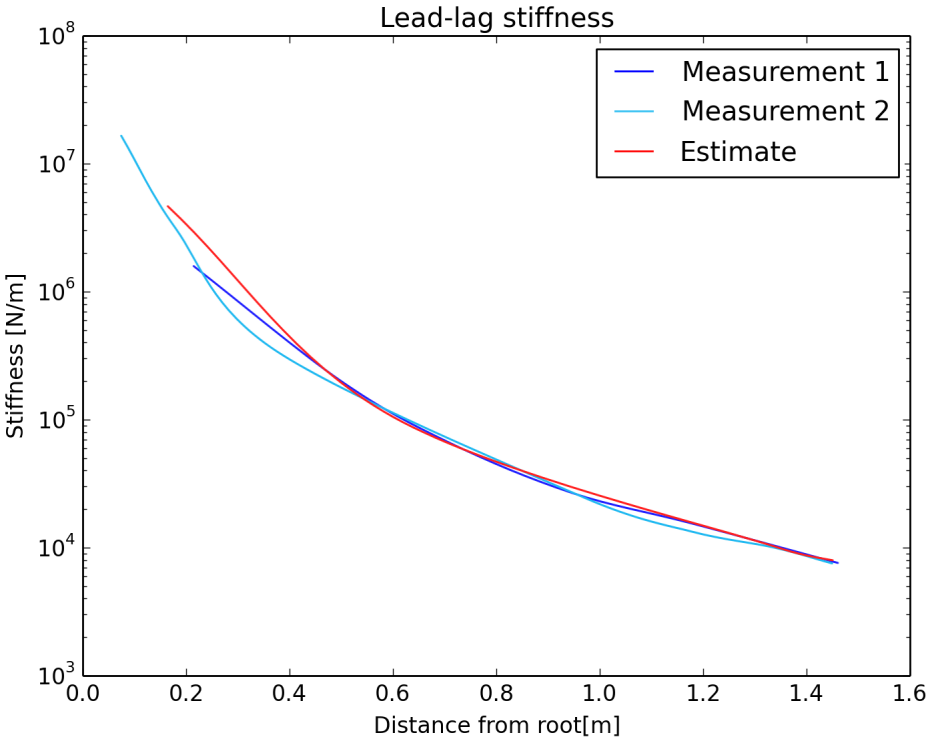


Figure 4.5: Lead-lag bending stiffness

The stiffness in x direction AE_x was difficult to measure. The blade is very stiff in that direction, so either very large forces or very accurate measurements were needed. Therefore the value of AE_x is approximated

using data from a pull out test performed at the TU Delft. From this data a mean value for AE_x could be found and it was assumed to be constant over the rotor blade.

The position of the neutral line y_{np} was calculated by the NLR and calculated using the results form the static deflection measurements. These last results were very inconsistent, therefore the results from NLR were used. The NLR determined the position of the neutral line at 80% of the span to be at 38mm from the leading edge and constant over the length of the rotor blade.

4.3. INERTIA PROPERTIES

The aeroelastic model needs the following inertia properties:

m_n The translational inertia (mass) of the element.
ρI_p The rotational inertia of the element.
y_{cg} y Position of centre of gravity.
z_{cg} z Position of centre of gravity.

The mass and position of centre of gravity were measured using three scales. The rotor blade is fixed on a rig. This rig has three contacts points with the ground. A scale is placed under each contact point. This measurement setup is shown in figure 4.6. The result of these measurements can be found in table 4.1.

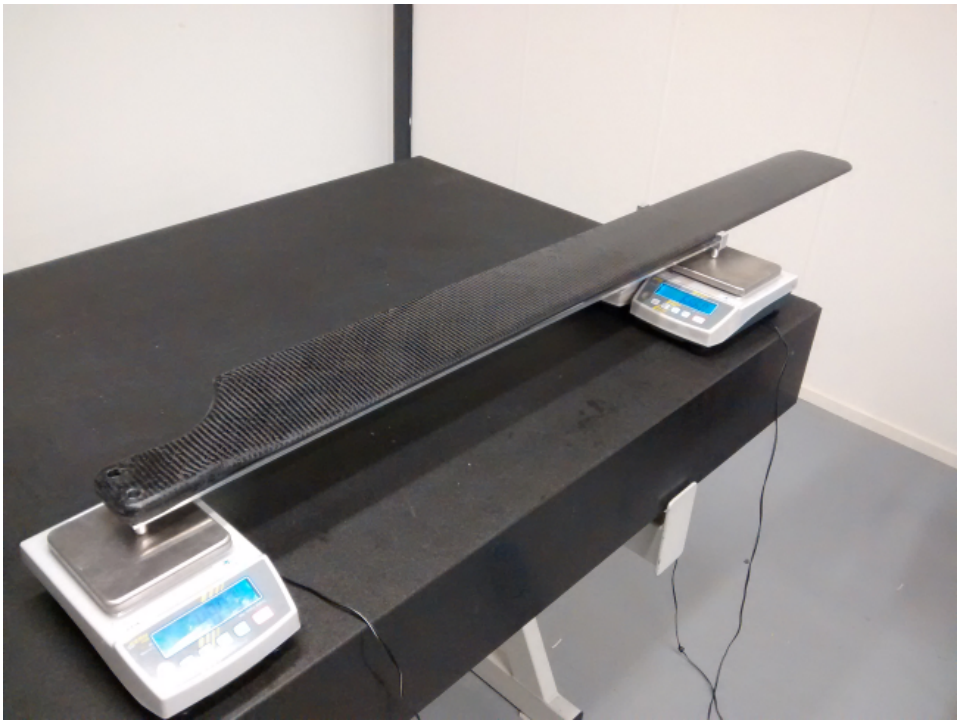


Figure 4.6: Determination of mass and position of centre of gravity

Blade	Weight$[g]$	$cg_x[mm]$	$cg_y[mm]$
P08	2129.0	715.6	44.7
P14	2071.1	753.7	42.9
P14	2062.3	754.0	42.9
P14	2062.7	754.5	42.8
P15	2069.4	754.2	43.1
P15	2073.6	746.8	43.3
P19	2116.0	747.3	43.7
P19	2116.3	747.2	43.7
P20	2117.0	747.5	43.8
P20	2117.5	747.6	43.7
P21	2106.3	741.0	43.9
P21	2100.1	744.3	43.7
P21	2100.0	744.3	43.8

Table 4.1: Centre of gravity for several blades.

Table 4.1 shows that some small differences in measured weight and cg-positions between blades and even between different measurements of the same blade. Therefore the average values for the mass and cg-positions are used, combined with data from the CAD-model to estimate the mass distribution and rotational inertia properties.

4.4. TESTING AND MEASUREMENTS

Several tests and measurements were be performed to verify (parts of) the model. This section describes these tests, the results of these tests can be found in chapter 5.

A static deflection measurement as described by Jung et al [24] and by Anusonti-Inthra and Liu [25] will be performed. In these tests the root of the blade is fixed. The blade is deformed by applying a load. These deflections were compared to those predicted by the model. This comparison measures if the stiffness of the blade is modelled correctly.

The natural frequencies are important parameters for any dynamically loaded structure. Therefore it is important verify how accurate the natural frequencies are predicted by the model [10]. To measure the natural frequencies the root of the blade was fixed, similar as with the static deflection measurement. The blade was tapped and the resulting movement was observed. These movements represent the mode shapes corresponding to the natural frequencies. The period of these movements was timed to determine the natural frequencies. These natural frequencies were compared to natural frequencies predicted by the aeroelastic model.

A series of whirl tower tests were performed to measure drag. The power required for rotation was measured at several angles of incidence and compared with values predicted by the model. Lift is more difficult to measure. It requires a load cell build in the whirl tower or a Pitot-tube rack to measure the downwash. This is beyond the scope of this thesis.

5

RESULTS AND DISCUSSION

The goal of this chapter is to verify (parts of) the aeroelastic model. The first four sections focus on the structural model. In section 5.1 and 5.2 the steady-state responses of a uniform beam and a rotor blade (none-uniform) are compared to theoretical responses. In section 5.3 the natural frequencies of a modelled uniform beam are compared to theoretical natural frequencies of a uniform beam. In section 5.4 the natural frequencies of a modelled rotor blade are compared to natural frequencies of the rotor blades measured by the NLR. In section 5.5 the aerodynamic model is tested: The aeroelastic models prediction of the power required for rotation is compared to the power required measured in whirl tower tests. Section 5.6 contains remarks and observations made during the tests described in the previous sections.

5.1. STATIC DEFLECTIONS OF AN UNIFORM BEAM

For this section a uniform beam was modelled. A load was applied at multiple points (one at a time) and the (steady-state) deflection predicted by the aeroelastic model was recorded. The applied load and the deflection were transformed into an equivalent stiffness of the beam from the root until that point. The results are shown in figures 5.1, 5.2 and 5.3 (blue lines). These results are compared with theoretical values (red lines).

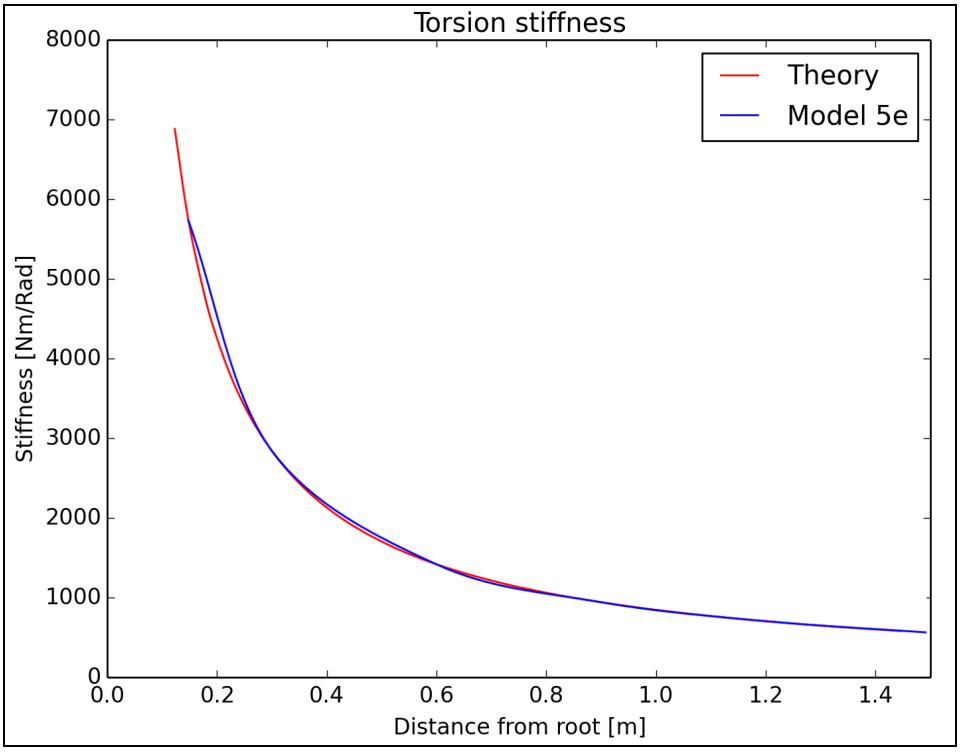


Figure 5.1: Torsion stiffness of a uniform beam.

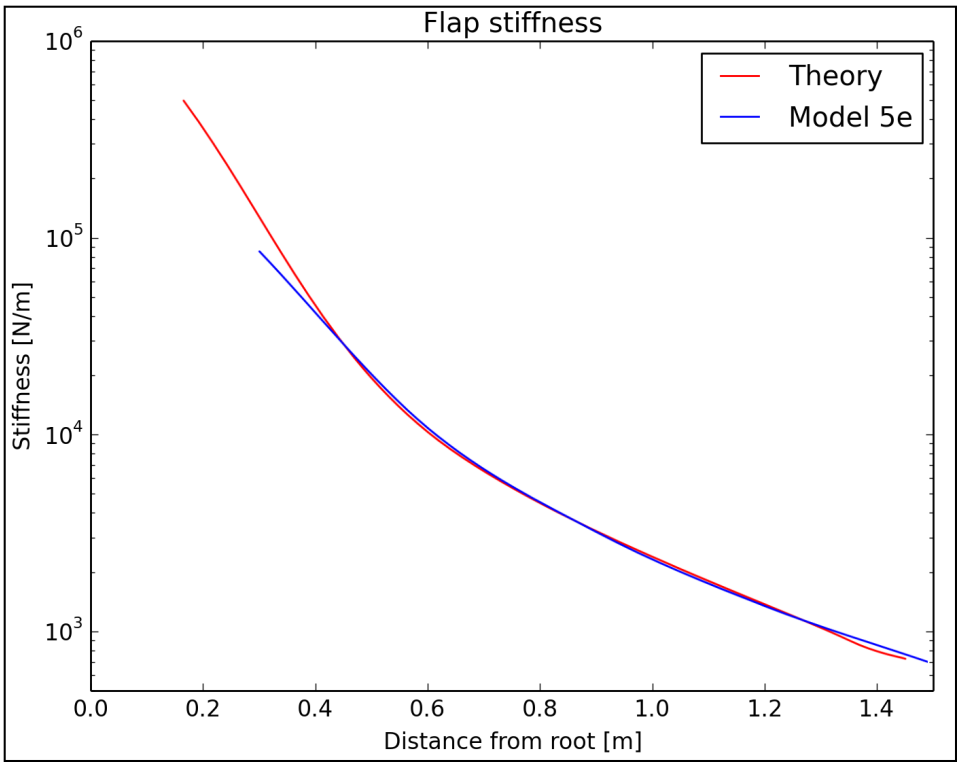


Figure 5.2: Flap bending stiffness of a uniform beam.

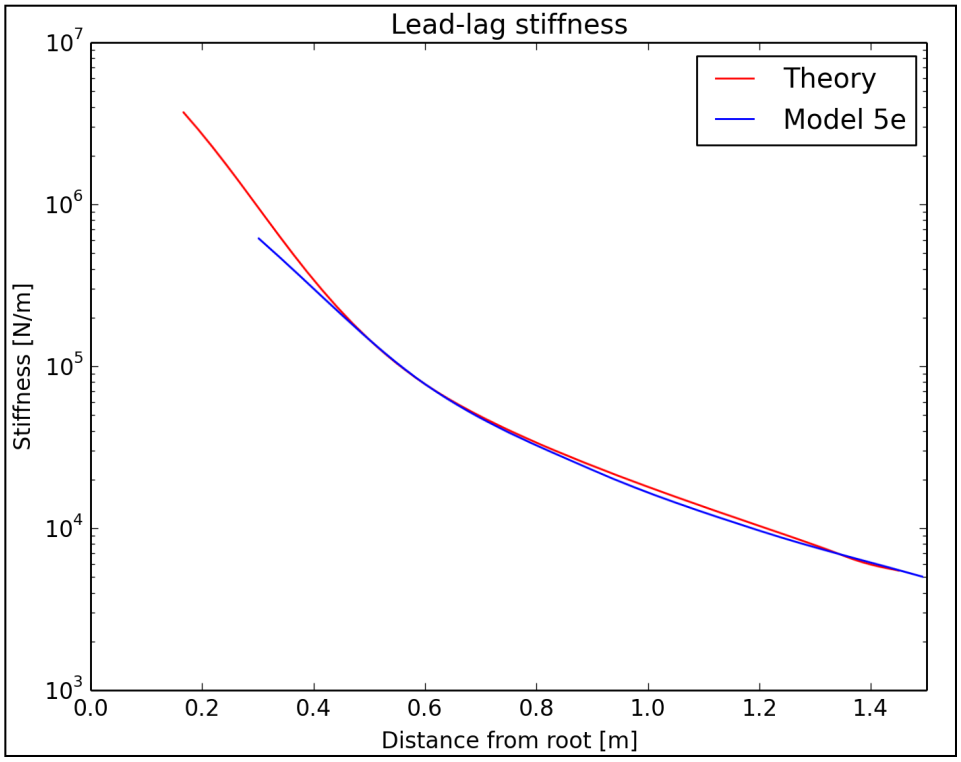


Figure 5.3: Lead-lag bending stiffness of a uniform beam.

Figures 5.1, 5.2 and 5.3 show that there is a good match between the results from the aeroelastic model and the results from theoretical calculations. This indicates (the steady state part of) the structural model worked correctly. If the correct stiffness values are used, the structural model will produce correct (steady state) results.

5.2. STATIC DEFLECTIONS OF A ROTOR BLADE

The test described in this section is similar to the test described in section 5.1, except for this test a rotor blade was modelled instead of an uniform beam. The KVE rotor blades have a non-uniform stiffness and mass distribution. Not only did this change the input stiffness values, it also introduced an error: In the aeroelastic model the stiffness and mass properties are assumed to be constant over an element. For an uniform beam this is a correct assumption, however for a non-uniform beam this is an approximation. The accuracy of this approximation depends on the number of elements used for the aeroelastic model. Therefore the test described in this section is also an investigation into the minimum number of elements needed to describe a rotor blade accurately.

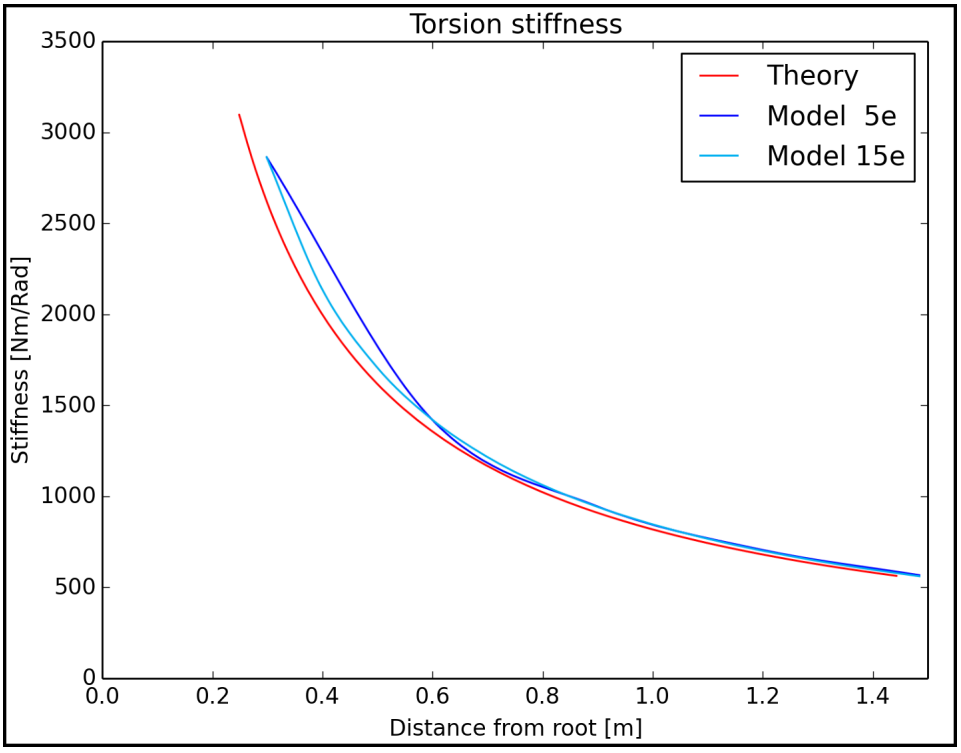


Figure 5.4: Torsion stiffness of a rotor blade.

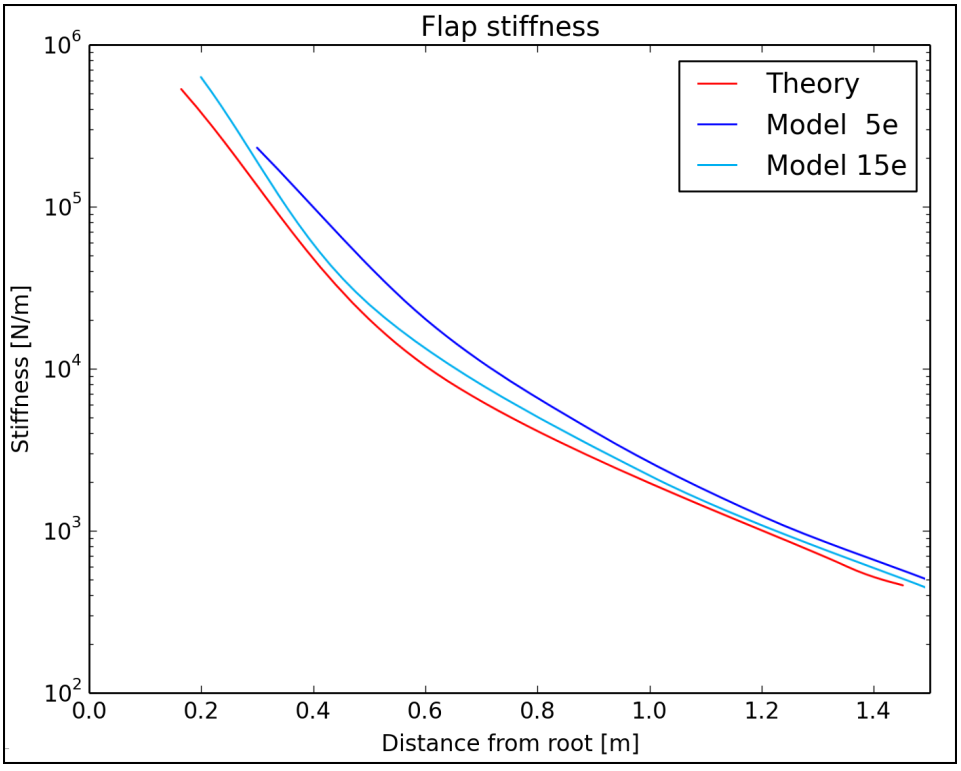


Figure 5.5: Flap bending stiffness of a rotor blade.

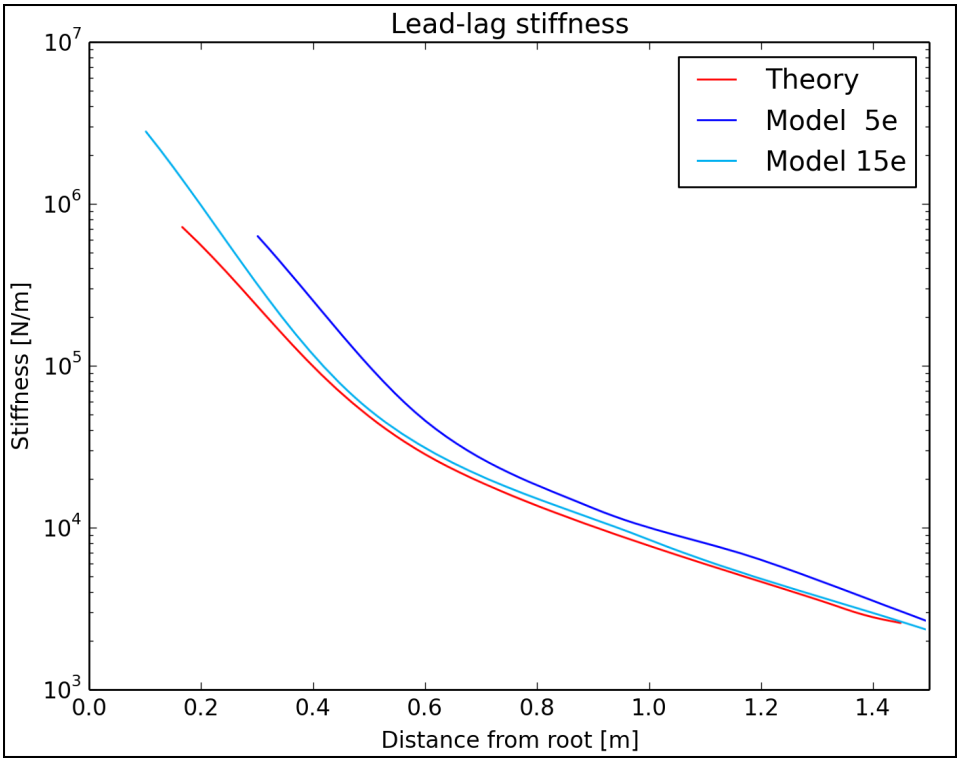


Figure 5.6: Lead-lag bending stiffness of a rotor blade.

Figures 5.4, 5.5 and 5.6 show that there is an error between the results from the aeroelastic model and the measurements. For the aeroelastic model using 5 elements the shape is similar, but the error is relatively large. For the aeroelastic model using 15 elements the error is greatly reduced.

5.3. NATURAL FREQUENCIES OF A UNIFORM BEAM

In this section a uniform beam was modelled similar as was done for section 5.1. In section 5.1 the response of the aeroelastic model was investigated, in this section the Eigenfrequencies of the aeroelastic model were investigated. The natural frequencies were calculated using the Campbell script described in section 3.4.2. These modelled natural frequencies were compared with the natural frequencies of a uniform beam given by literature [8, 10, 26].

For comparison of the natural frequencies an error function was defined:

$$error = \frac{\omega_{literature} - \omega_{model}}{\omega_{literature}} \tag{5.1}$$

The error values are plotted against the number of elements used to model the uniform beam for all motions in figure 5.7.

These results show that with only 8 elements an accurate prediction can be made of the first two natural frequencies for all motions. The predictions of the third natural frequencies for the y- and z- motion are also very accurate, while the prediction for the x- and torsion-motion requires a few more elements.

5.4. NATURAL FREQUENCIES OF A ROTOR BLADE

For this section the natural frequencies of a (non-uniform) non-rotating rotor blade was investigated. The natural frequencies of three blades were measured by the NLR. These results are displayed in table 5.1 as blade 1, blade 2 and blade 3. These measured natural frequencies were compared to the natural frequencies predicted by the aeroelastic model. These results are shown in table 5.1 as calculated followed by the number of elements used.

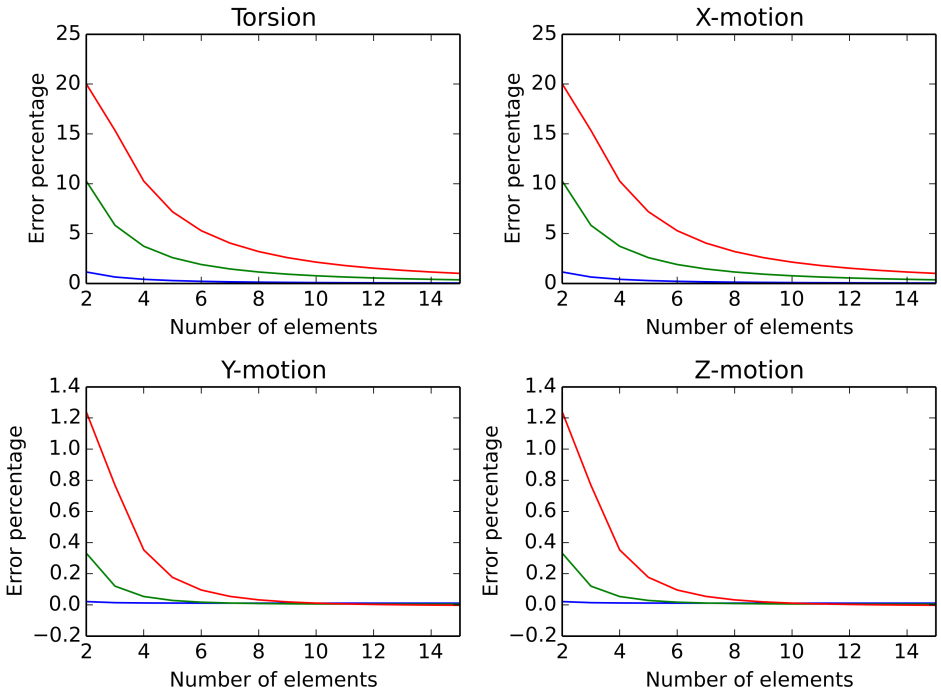


Figure 5.7: Error percentage of the calculated first (blue line), second (green line) and third (red line) natural frequency for the four sets of equations.

Method	1st torsion	1st bending	2nd bending	3rd bending	lead-lag
Measurement 1	106.75	5.10	30.56	78.65	18.87
Measurement 2	112.02	5.08	31.59	84.69	20.09
Measurement 3	113.75	5.32	32.56	85.59	21.69
Calculated 5e	127.51	7.03	42.00	107.79	33.59
Calculated 15e	114.03	5.52	33.61	96.31	22.89
Calculated 30e	111.98	5.25	31.59	91.80	21.77
Calculated 150e	111.64	5.09	30.40	88.88	20.06
Calculated 300e	111.82	5.08	30.29	88.49	19.90

Table 5.1: Natural frequencies [Hz], NLR measurements of three blades (measurement 1 to 3) and calculated (calculated) using different number of elements.

The values in table 5.1 show that the aeroelastic model using 5 elements is not capable of predicting the natural frequencies of the rotor blade accurately. If the aerolastic model uses 15 elements the calculated natural frequency for the first torsion mode, the first two flap bending modes and the first lead-lag bending mode match the measured natural frequencies quite well. For the third bending flap bending mode to be accurate an aeroelastic model using 150 elements is required.

5.5. POWER CURVE

The power required for rotating a blade is measured on KVE's whirl tower. An estimate of the efficiency curve (efficiency vs rpm) for the drive train (motor and chain transmission) was made based on literature [27–30]. Using the voltage and current measured by the inverter and the efficiency curve for the drive train, the power required for the rotation of a rotor blade was derived. The power required derived from the whirl tower tests was compared to the power required predicted by the aeroelastic model using both inflow models. This comparison is shown in figure 5.8.

Note: KVE has plans to build a new whirl tower. The design of this new whirl tower features a direct-co-axial drive train. This reduces the range of rotational speeds for the whirl tower, but it also eliminates the chain transmission. With this new whirl tower the power required to rotate the blade could be measured more

accurately. In this direct-co-axial drive train design the motor axle and the whirl tower axle are connected by a flexible coupling part, to allow for misalignment of the axles. One of the flexible coupling parts considered can measure its rotation speed and the applied torque. This would allow for an even more accurate determination of the power required.

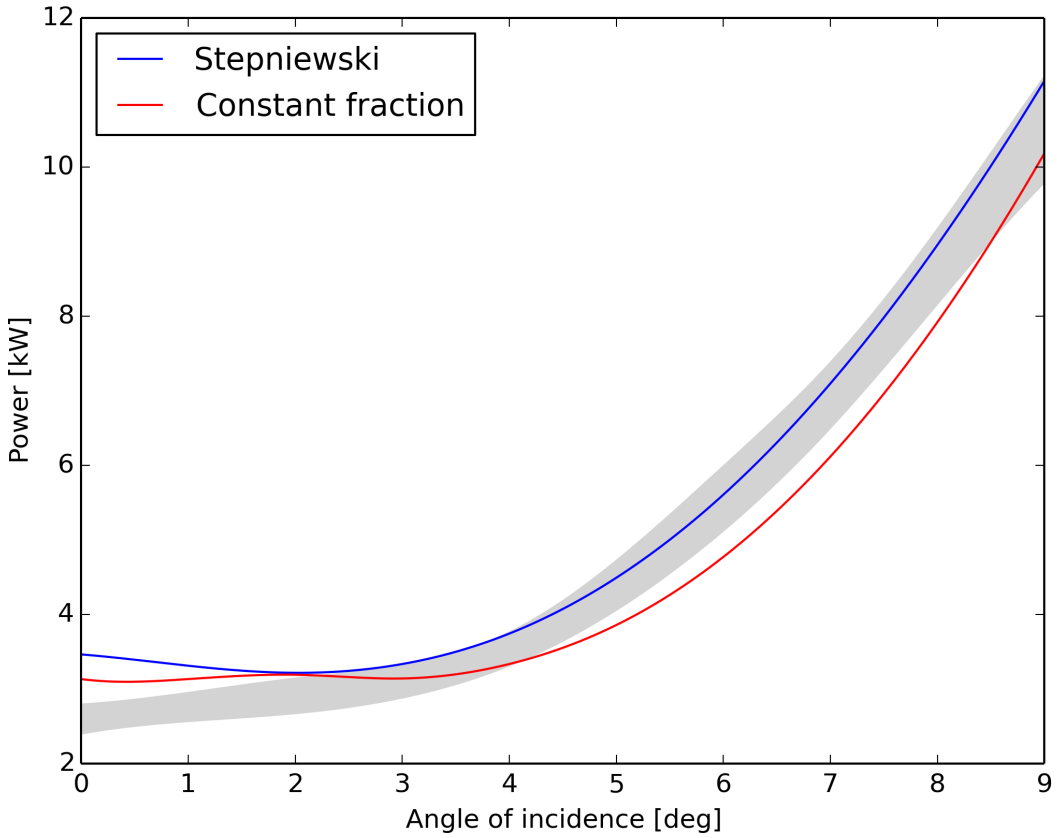


Figure 5.8: Range of measured aerodynamic power during whirl tower tests (grey area) and aerodynamic power calculated by the aeroelastic model using both inlfow models (red and blue lines).

The power curves predicted by the aeroelastic model, shown in figure 5.8, have a similar shape as the power curves obtained during whirl tower tests, for both inflow models. However the power required predicted by the aeroelastic model at low collective angles is considerably higher than the power required values measured, for both inflow models. This over estimation at low collective angles occurs when the blade produces both negative and positive values for v_i (due to the twist distribution). It seems that both inflow models are not capable of predicting the induced velocities at low collective angles correctly. At high collective angles the prediction of the aeroelastic model using Stepniewski's inflow model is within the values measured. The prediction of the aeroelastic model using the constant fraction inflow model seems to be slightly below the power required measured.

5.6. OTHER OBSERVATIONS

This section contains observations and remarks on whether or not the model agrees with measurements. These observations were made during the tests described in the previous sections.

5.6.1. THRUST

Figure 5.8 showed that the aeroelastic models prediction of power required using Stepniewski's inflow model was slightly more accurate compared to the aeroelastic models prediction using the constant fraction inflow model, at high collective angles. The thrust was not measured, but Adasi stated that a collective of 7° and

1186 rpm is what they call "hover settings". The UAV has a maximum take off weight of 200kg and it has two rotor blades. This means that at ”hoover settings” there is approximately 100kg thrust per blade. The aeroelastic model using Stepniewski's inflow model predicts at these "hover settings” around $1400N$ thrust while the aeroelastic model using the ”constant fraction inflow model predicts $1265N$ thrust. Both values seem too high, but the predicted thrust value of the aeroelastic model using the constant fraction inflow model is not unlikely when considering ground effects. The thrust prediction of the aeroelastic model using the Stepniewski inflow model seems to be too high even when considering ground effects.

5.6.2. MAXIMUM POWER

Adasi also stated that the engine in their UAV does not have enough power for the UAV to maintain a 9° collective angle at nominal rotational speed. The engine of their UAV is rated at 37kW. This is approximately 13-15kW per blade, assuming 10% mechanical losses and 10%-20% power consumption by the tail rotor [16, 31]. Figure 5.8 shows that the aeroelastic models predictions of the required power at these settings is at 10.5-11kW depending on the inflow model used and during the whirl tower tests these values were between 9.5-10.5kW. So there is a significant difference in power required between the values based on the rated power of the engine and the values obtained in whirl tower test and aeroelastic model predictions. This can be caused by a much lower power output of the engine, different mechanical losses or if more power is consumed by the tail rotor. The actual power output can be different from the rated power, due to high ambient temperatures in UAE.

5.6.3. INFLUENCE OF ROTOR BLADE TWIST DISTRIBUTION

KVE had produced a few sets of rotor blades that seemed similar to the original rotor blades at first glance. However when these rotor blades were tested on the whirl tower KVE found that the whirl tower required significant more power for spinning these new rotor blades than it required for spinning the original rotor blades. An investigation into this power difference was started and a small difference in the twist distribution was detected. The NLR investigated the influence of these differences in twist distribution. They concluded that a change in twist would cause a horizontal shift in the power curve. This investigation was repeated using the aeroelastic model. The result is shown in figure 5.9. It shows two power curves created using the aeroelastic model (constant fraction inflow model). The blue line is a power curve for a rotor blade that has the standard twist distribution $(5°, -2°)$, the red line is a power curve for a rotor blade that has a slightly altered twist distribution $(5°, -1.5°)$. The difference between these two power curves is in accordance with NLR's findings.

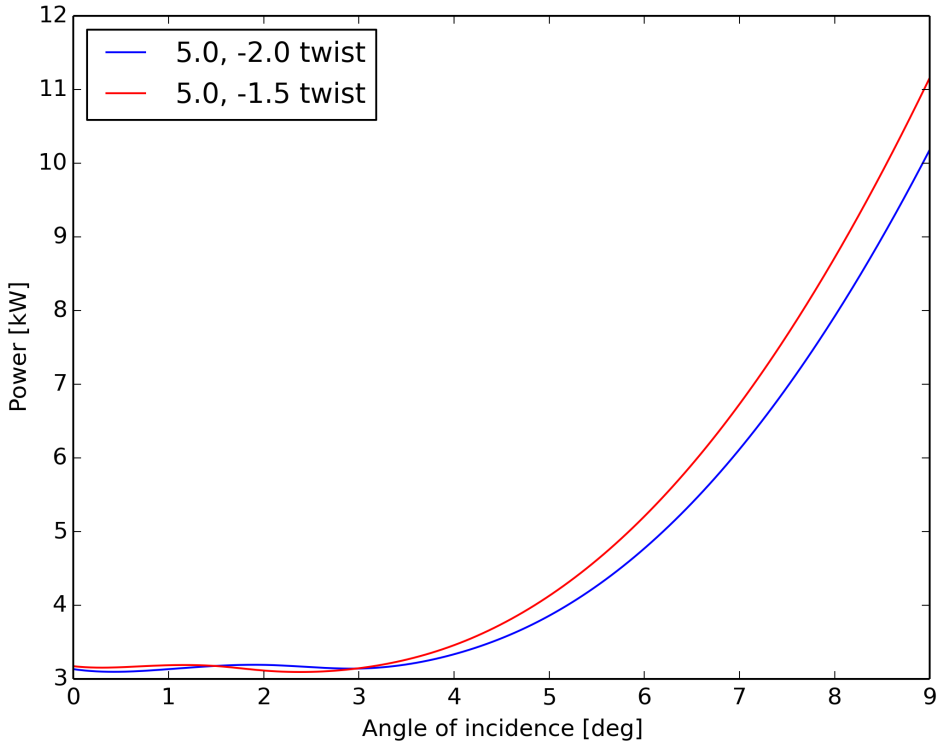


Figure 5.9: Aerodynamic power required for blades with a $5°, -2°$ twist distribution (blue line) and with a $5°, -1.5°$ twist distribution (red line)

5.6.4. FLAP DISPLACEMENT

In this section the flap (bending) displacement at the tip is discussed. The aeroelastic model predicts a slightly negative to zero flap displacement at $0°$ collective, a $13-14cm$ flap displacement at $7°$ and a $17-18cm$ flap displacement at $9°$, using either of the two discussed inflow models. The flap displacement was never measured during whirl tower tests, however an estimate can be made using the images and videos of the whirl tower test. The cover picture is one example of a picture taken during whirl tower testing. Figure 5.10 is constructed out of two ($0°$ and $7°$ collective) images made during whirl tower testing. It is very difficult to obtain accurate numbers from these images because of the absence of a good reference, but some estimations can be done: the rotor blade had a slightly negative flap displacement at $0°$ collective angle and approximately a $15cm$ displacement at $7°$ collective. This is in accordance with predictions from the model.

5.6.5. INFLUENCE OF ROTOR BLADE STIFFNESS

Calculations by the NLR showed that the blade bending stiffness has a negligible influence on aerodynamic performance, because most of the rotor blades stiffness is caused by the stabilizing effect of the centripetal force. Figure 5.11 shows the power curve predicted by the aeroelastic model of a infinite stiff (rigid blade (blue line) and a standard rotor blade (red line). These results are in accordance with in the findings of the NLR.

Figure 5.10: Blade deflection at 0° and at 7° collective

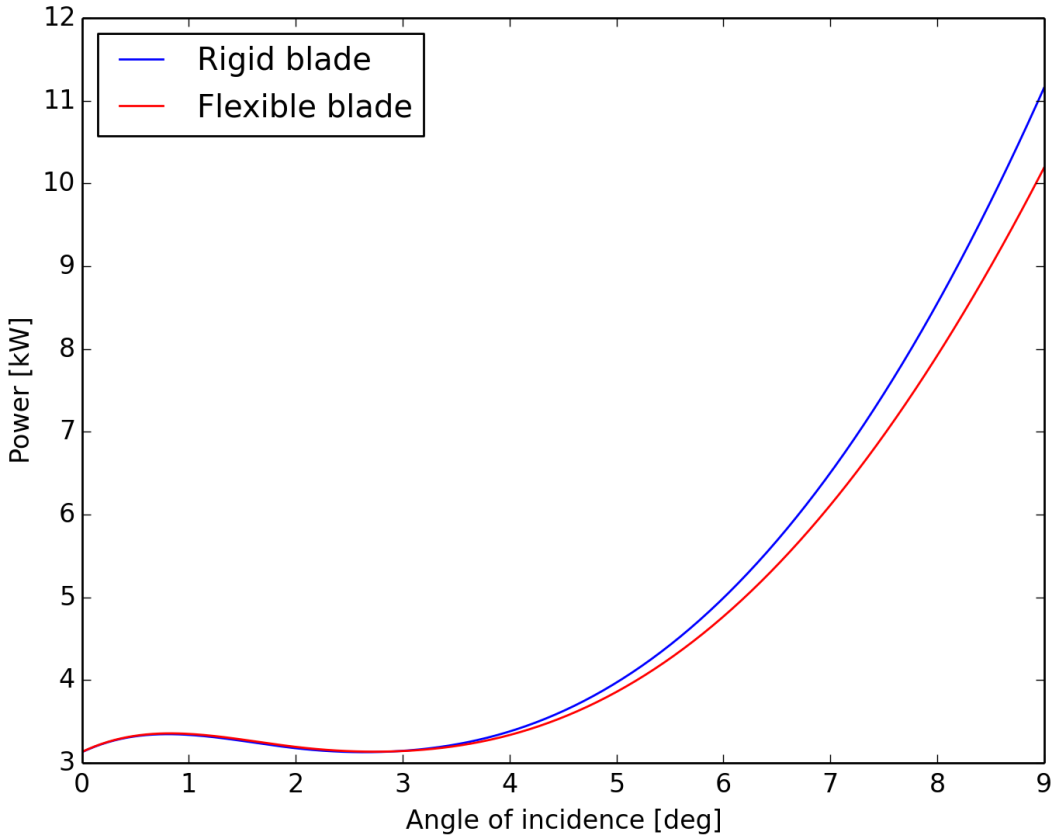


Figure 5.11: Aerodynamic response of a standard blade (blue line) and a blade with a 10% increased stiffness (red line) at 7◦ collective

6

CONCLUSIONS AND RECOMMENDATIONS

6.1. CONCLUSIONS

In this report the development of an an aeroelastic model was described. The aim of this aeroelastic model is to provide KVE a tool for the design phase of a rotor blade. Using this aeroelastic model KVE should be able to:

- Determine the static and dynamic deformations of a rotor blade during flight or whirl tower test.
- Determine the static and dynamic forces acting on a rotor blade during flight or whirl tower test.
- Determine the natural frequencies of the blade at different settings (rotational speed, pitch, etc.).
- Evaluate the consequences of design changes on the stability, deformations and stresses.
- Compare the aeroelastic behaviour of different rotor blade designs.

An aeroelastic was created consisting of a Ritz-Hamilton structural model, a lifting line quasi steady aerodynamic model with two inflow models. The aeroelastic model includes effects due to centripetal and gravity forces and conservation of angular momentum. A fourth order Runge-Kutta integration method is used to numerically integrate the aeroelastic model to obtain the response of the system.

The aeroelastic model was validated by comparing calculated results with theoretical results and results obtained by measurements. Both a uniform beam and a rotor blade were modelled. The calculated steady state deflections of the uniform beam were compared with values obtained from literature. The calculated steady state deflections of the rotor blades were compared with values obtained from measurements. The calculated natural frequencies for a uniform beam compared to values obtained from literature. The calculated natural frequencies for a rotor blade were compared with values obtained by measurements done by the NLR. The calculated drag for the rotor blades was transformed into a power curve and compared with values obtained in a series of whirl tower tests.

The comparison of the steady state deflection of a uniform beam showed a match between the results from the aeroelastic model and the results from theoretical calculations. This indicates (the steady state part of) the structural model worked correctly. If the correct stiffness values are used, the structural model will produce correct (steady state) results.

The comparison of the steady state deflection of a rotor bladed showed that there is some error between the results from the aeroelastic model and the measurements. For the aeroelastic model using 5 elements the shape is similar, but the error is relatively large and for the aeroelastic model using 15 elements the error is greatly reduced.

The comparison of the natural frequencies for a uniform beam showed that with only 8 elements an accurate prediction can be made of the first two natural frequencies for all motions. The predictions of the third natural frequencies for the y- and z- motion were also very accurate, while the prediction for the x- and torsion-motion required a few more elements.

The comparison of the natural frequencies for a rotor blade showed that the aeroelastic model using only 5 elements is not capable of predicting the natural frequencies of the rotor blade accurately. If the aeroelastic model uses 15 elements the calculated natural frequency for the first torsion mode, the first two flap bending modes and the first lead-lag bending mode match the measured natural frequencies quite well. For the third bending flap bending mode to be accurate an aeroelastic model using 150 elements is required.

The power curves predicted by the aeroelastic model in figure 5.8 have a similar shape as the power curves through the points measured, for both the inflow models. The power required predicted by the aeroelastic model at low collective angles is slightly higher than the power required values measured. This over estimation at low collective angles occurs when the blade produces both negative and positive values for v_i (due to the twist distribution). It seems that both inflow models are not capable of predicting this correctly. At high collective angles the prediction of the aeroelastic model using Stepniewski's inflow model is within the values measured. The prediction of the aeroelastic model using the constant fraction inflow model seems to be slightly below the power required measured.

The steady state response and natural frequencies of a uniform beam are very close the values from literature. The error for the steady state response and natural frequencies of a rotor blade is larger than that for the uniform beam. This can be explained by the uniform beam representation of each section. This error reduces if more elements are used. The minimum elements to use depends on the variable of interest. The power predicted is higher than measured at low angels of incidence and within the bounds measured for high angles of incidence. Using this model KVE is able to predict initial loads, deflections and the consequences of changes in the design of their rotor blade. However further validation of the model is recommended.

Between the last measurement for this report and the finalization of this report KVE has successfully used the aeroelastic model to develop a new rotor blade. A requirement for this new project was a limited maximum deflection. The deflection at maximum angle of incidence of the design was well above this limit. However this maximum angle of incidence can not be reached while hovering. This requires a lot more power than the engine can provide. The maximum angle of incidence can only be reached in forward flight, when the swashplate is set at an angle and the maximum angle of incidence is alternated with a much lower angle of incidence. The aeroelastic models results showed that while the blade reached it's maximum angle of incidence the blade did not have enough time to bend into the corresponding deflection: The deflections remained under the specified limit.

6.2. RECOMMENDATIONS

On account of the work described in this report the following recommendations are made:

- Perform power curve measurements on the new whirl tower.
- Determine motor and drive train efficiency more accurately
- Perform a lift curve measurement.
- Strain gauge measurement to verify displacements and loads calculations
- Determine the rotor blades inertia properties more accurately.

There are plans to build a new direct-drive whirl tower. The elimination of the chain transmission and the (optional) torque measuring coupling between the engine and main axle would allow the power required to be measured more accurately. Alternatively the efficiency of the motor and drive train can be measured

It would be interesting to compare the lift predicted by the aeroelastic model with the actual lift. More accurate power curves and a lift curves are recommended to to verify the inflow models and aerodynamic model further. However lift is more difficult to measure than drag. An attempt was made to measure lift, using an improvised pitot rack setup. The results were not as expected and the project was abandon due to lack of time. A more direct method would be placing the whirl tower or the actual helicopter on a large scale.

On of the goals of this thesis was to provide KVE a tool to determine the loads on a rotor blade. Therefore it is recommended to compare the largest load on a rotor blade: the centripetal force. This can be done by measuring strain using strain gauges during a whirl tower test and compare the results with strains predicted

by the aeroelastic model. Strain gauge measurements could also help determining the lift by comparing the strain in the upper and lower side of the blade.

Another intended use for the aeroelastic model is to use it for natural frequency analysis. Crucial in determining the natural frequencies is obtaining the correct stiffness and inertia properties. There is a room for improvement in determining both, but especially for the inertia properties. An option would be to saw a blade in pieces and determine the weight and cg position of each section individually.

BIBLIOGRAPHY

[1] W. Stepniewski and C. N. Keys, *Rotary-wing aerodynamics. Volume 1: Basic theories of rotor aerodynamics with application to helicopters* (NASA, 1979).

[2] S. Shams, M. H. Sadr, and H. Haddadpour, *An efficient method for nonlinear aeroelasticy of slender wings* (2011).

[3] S. Y. Wie, D. K. Im, E. Kim, J. H. Kwon, and D. J. Lee, *An Analysis on the Helicopter Rotor Aerodynamics in Hover and Forward Flight Using CFD/Time-Marching-Free-Wake Coupling Method* (2009).

[4] D. M. Schuster, D. D. Liu, and L. J. Huttsell, *Computational Aeroelasticity: Success, Progress, Challenge* (2003).

[5] E. Livne, *Future of Airplane Aeroelasticity* (2003).

[6] K. Lai and T. Kim, *Reduced-Order Aeroelastic Modeling Using Coupled CFD-CSD Simulations and System Identification Technique* (2013).

[7] J. G. Leishman, *Rotorcraft Aerodynamics* (2010).

[8] Piersol, Allan G. and Paez, Thomas L., *Harris' Shock and Vibration Handbook* (1996).

[9] E. H. Dowell, R. Clark, D. Cox, J. H.C. Curtiss, J. W. Edwards, K. C. Hall, D. A. Peters, R. Scanlan, E. Simiu, F. Sisto, and T. W. Strganac, *A Modern Course in Aeroelasticity,* (1981).

[10] D. J. Inman, *Engineering Vibration* (2001).

[11] J. Meriam and L. Kraige, *Engineering Mechanics: Dynamics*, Engineering Mechanics (Wiley, 1998).

[12] J. S. Torok, *Analytical Mechanics: With an Introduction to Dynamical Systems* (Wiley, 1999).

[13] J. W. Eischen, *Matrix Analysis of Beams* (1991).

[14] H. B. SEON M. HAN and T. WEI, *Dynamics of Transversely Vibrating Beams Using Four Engineering Theories* (Academic Press, 1999).

[15] J. M. Gere and S. P. Timoshenko, *Mechanics of Materials* (Nelson Thornes Limited, 1999).

[16] J. Leishman, *Principles of Helicopter Aerodynamics* (Cambridge University Press, 2006).

[17] J. Anderson, *Fundamentals of aerodynamics*, McGraw-Hill series in aeronautical and aerospace engineering (McGraw-Hill, 2001).

[18] G. Ruijgrok, *Elements of airplane performance* (Centraal Boekhuis, 1990).

[19] T. Holten and J. Melkert, *Helicopter Performance, Stability and Control* (2002).

[20] J. Anderson, *Introduction to Flight* (McGraw-Hill, 2000).

[21] R. Szczepanik, R. Przysowa, J. Spychala, J. Rokicki, K. Kaźmierczak, and P. PMajewski, *Application of blade-tip sensors to blade-vibration monitoring in gas turbines,* in *Thermal Power Plants* (Intech, 2012).

[22] G. Segal, F. Vermolen, and G. van Kan, *Numerical Methods in Scientific Computing* (2005).

[23] M. Drela and H. Youngren, *Xfoil 6.99* (2013).

[24] S. N. Jung, Y. H. You, B. H. Lau, W. Johnson, and J. W. Lim, *Evaluation of rotor structural and aerodynamic loads using measured blade properties,* Journal of the American Helicopter Society **58**, 1 (2013).

[25] P. Anusonti-Inthra and Y. Liu, *Structural Property Determination for an Articulated Rotor Blade* (National Institute of Aerospace, 2006).

[26] T. H. G. Megson, *Aircraft Structures for Engineering Students* (Edward Arnold, 1999).

[27] *Determining Electric Motor Load and Efficiency* (U.S. Department of Energy).

[28] T. Jones, *Motor Efficiency, Selection, and Management, A Guidebook for Industrial Efficiency Programs* (Consortium for Energy Efficiency, 2013).

[29] R. Schiferl, *An Accurate Method to Determine Electric Motor Efficiency While the Motor is in Operation* (Rockwell Automation, 2003).

[30] P. Oberoi and S. K. Aggarwal, *Electrical Energy Auditing and Harmonic Analysis of Industrial Units: A Case Study* (2011).

[31] W. Wieser, *Tail Rotor Design Guide* (1974).

A

APPENDIX: GETTING STARTED

The model is written in Python, a free, open-source and easy to learn program language.

A.1. DOWNLOAD AND "INSTALLATION"

The following instructions will use WinPython. This is a portable package for Windows based systems. It contains the Python runtime and Spyder. Spyder can be used as an editior of Python files and as an interface for the Python interpreter (Python command prompt).

Setting up WinPython consists of the following steps:

1. Download WinPython. (During this project Python 2.7 was used)
2. Run the downloaded executable. This will extract WinPython to a folder of choice.
3. Run the Spyder executable located in the extraction folder.

It is possible to install Python and Spyder separately on Linux and Mac. It is also possible to install Python and use a different interface and editor. Instructions and documentation for this is widely available on the internet, but will not be discussed in this section.

A.2. SPYDER INTERFACE

Figure A.1 shows Spyder's default interface. It shows much resemblance with Matlab's interface. The red lines are added to highlight four area's in the interface: The button and menu area (top), the editor (left), the variable explorer/object inspector (middle-right) and the Python interpreter (bottom-right).

The button and menu area contains the buttons for opening, saving, creating new and running files.

The editor displays the code of the loadedf python scripts. Python files can be drag and dropped or opened and saved using the button and menu area. Multiple files are shown in a tabbed structure.

The variable explorer/object inspector can be switched to display the variable explorer or the object explorer by the corresponding buttons at the lower side. The variable explorer displays the variables in the memory of the Python interpreter. This can be helpful when trouble shooting. For example: if the Python interpreter gives an dimensions-miss-match-error while plotting variables, the variable explorer can be used to inspect the dimensions. The object explorer acts as an help interface. It provides information about functions and packages.

The Python interpreter is the Python command prompt. It can be used to type code like "x=3". It can be used to get the value of a variable. For example typing "x" and enter will return the value of x. It can also be used as an output for a script. For example if a script contains the line "print x" the Python interpreter will

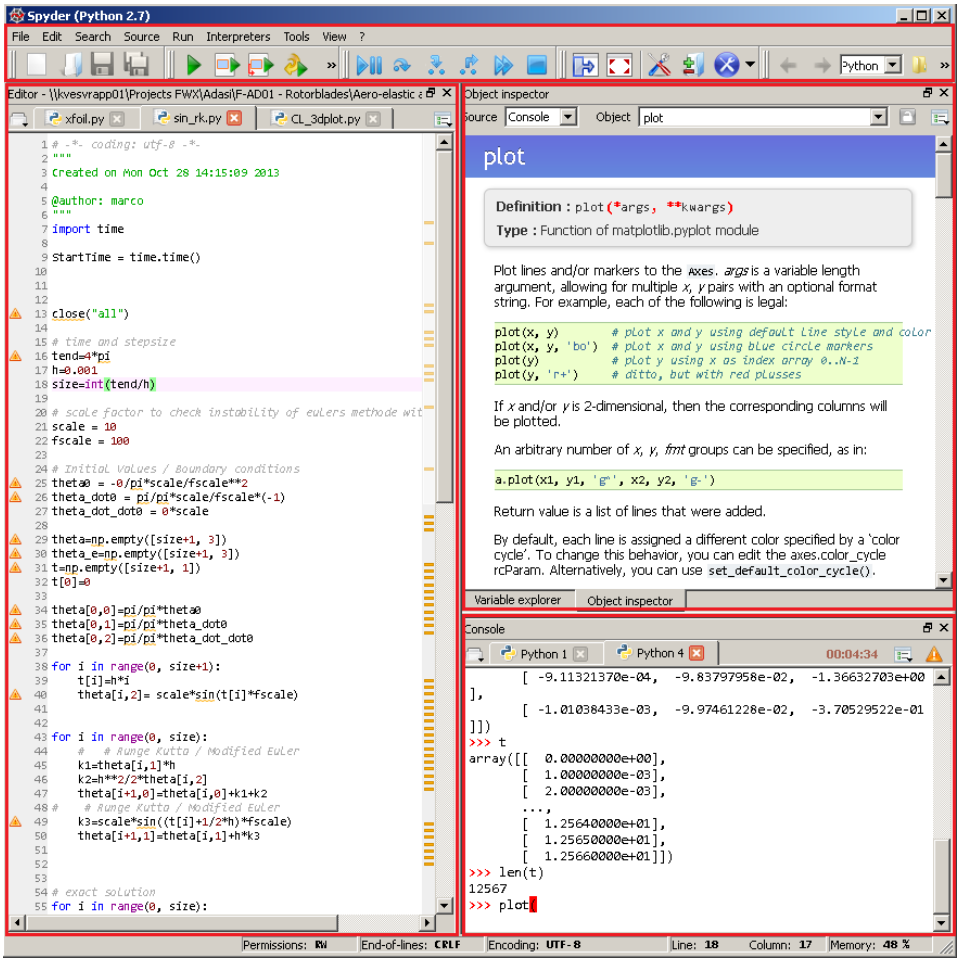


Figure A.1: Spyders interface with parts marked by red rectangles

display the value of x when the script runs. It can also be used to start a script. Like the Windows command prompt it is possible to browse through previous used commands using the arrows up and down. Multiple Python interpreters are displayed in a tabbed structure and run independent of each other. This means you can run two programs assigning different values to the same variables without affecting each other.

A.3. PROGRAMMING

Python scripts have a "py" extension and can be opened, changed and saved using Notepad, Spyder or a similar program. A Python script can contain values, code and functions and can import other scripts or packages. Figure A.2 shows two simple programs that import two other scripts or packages: antigravity and numpy. The figure also shows two methods to calculate the values for y_1 and y_2. The left program is a simple "line by line" calculation while the right program defines a function which is called later in line 22. The result, the figure the two programs produce are identical.

Note: numpy is imported in the Python interpreter by WinPython by default so that line can be left out. However functions run independent of the interpreter that calls them and for those the line needs to be included if the function uses numpy.

```
# -*- coding: utf-8 -*-
"""
Created on Wed Aug 13 09:38:16 2014

@author: marco
"""

import antigravity
import numpy as np

# time
t=np.linspace(0, 2*np.pi, num=100)









# Y functions
Y1 = sin(t)
Y2 = cos(t)

# Plotting
figure(1)
plot(t,Y1)
plot(t,Y2)
xlabel('Time [s]')
ylabel('Y [m]')
```

```
# -*- coding: utf-8 -*-
"""
Created on Wed Aug 13 09:38:16 2014

@author: marco
"""

import antigravity
import numpy as np

# time
t=np.linspace(0, 2*np.pi, num=100)

# Define function
def somefunction(s):
    "Instructions"
    X1 = sin(s)
    X2 = cos(s)
    return [X1, X2]

# Y functions
Y1, Y2 = somefunction(t)


# Plotting
figure(1)
plot(t,Y1)
plot(t,Y2)
xlabel('Time [s]')
ylabel('Y [m]')
```

Figure A.2: Two simple programs

B

APPENDIX: OVERVIEW OF FILES AND FUNCTIONS

This chapter provides a list of files and a list of functions. This section can be used to locate a file or function.

B.1. FILES

The model uses of the following files:

File	Folder	Description
Aero_Model.py	\	Contains aerodynamic functions
airfoil.dat	\Xfoil	File containing airfoil shape.
BC_f.py	\	File containing a function that returns boundary conditions.
Campbell.py	\	Script to calculate and plot the natural frequencies.
Forces_Moldel.py	\	File containing forces functions.
inputfile.py	\	File containing parameters.
inputfile2.py	\	File containing parameters, used by Response2.py.
input_aero.csv*	\Data*	File containing twist and chord length data
input_struc.csv*	\Data*	File containing structural data data
input_swash.csv*	\Data*	File containing description of swashplate movement
Integrate_f	\	File containing integration function.
Response.py	\	Script to calculate and plot the response of the blade.
Response2.py	\	Script similar to Response.py, but it can use different step sizes.
Structural_Model.py	\	File containing structural model functions.
test.py	\	Script to test functions separately.
Undeformed performance.py	\	Script to calculate performance of rigid blade.
Xfoil_files.log**	\Data\Xfoil	Set of Xfoil data files.
Xfoil.exe	\Xfoil	Xfoil program [23].
xfoil.py	\Xfoil	Script that uses Xfoil.exe to create a set of Xfoil data files.
xfoil_blend.py	\Xfoil	Script similar to xfoil.py but for blended airfoils.

Table B.1: Files

* Default name, extension or location defined by input file or program. Name and location can be changed.
** Name and extension are irrelevant as long as the files are in plain text.

Note that a distinguish is made between script and file. A file contains either one or more functions or data. Functions can not be executed, they can be called by a command prompt, script or an other function and return calculated values. A script is what is labelled "Commander" in the flowcharts: it can be executed, it can perform calculations, plot and call functions.

B.2. FUNCTIONS

The model consists of the following functions:

Function	Location	Description
Swashplate	Aero_Model.py	swashplate function.
aero_fo	Aero_Model.py	Function to create aerodynamic interpolation functions.
aero	Aero_Model.py	Aerodynamic performance function.
vi_f	Aero_Model.py	Inflow function: constant fraction model.
vi_f2	Aero_Model.py	Inflow function: Stepniewski inflow model.
BC	BC_f.py	Boundary condition input file.
Fcentripetal	Forces_Model.py	Function that models centripetal force.
Fmomentum	Forces_Model.py	Function that returns zeros.
Fmomentum2	Forces_Model.py	Function that models forces due to conservation of momentum.
Fgravity	Forces_Model.py	Function that models gravity force.
combine_F	Forces_Model.py	Function used by Response2.py.
expand_F	Forces_Model.py	Function used by Response2.py.
Integrate	Integrate_f.py	Integration function: mid-point method.
Integrate	Integrate_f2.py	Integration function: fourth order Runge-Kutta method.
FO_struc	Structural_Model.py	Function to interpolate input file to given number of nodes.
KMMatrix	Structural_Model.py	Function to calculate K- and M-matrices.

Table B.2: Functions

C

APPENDIX: INPUT FILES

C.1. INPUTFILE.PY

```
# Path and filenames of input files
file_struc = 'Data\input_struc_Adasi.csv' # Import file, structural data blade
file_aero = 'Data\input_aero.csv'
file_swash = 'Data\input_swash_15.csv'
xfoildf ='Data/Xfoil' #Path to folder containing xfoil data files

# Path of results folder
# If result_path_add =0 the results will be saved in \Results\[Time]\
# If result_path_add =1 the results will be saved in \Results\[result_path_addition]\[Time]\
result_path_add = 1
result_path_addition = 'test/'

# Number of plots
nplot=1000
nplot=10
# Nuber of notifications during calculation: 1/nnot complete, 2/nnot complete, etc
nnot = 10

# Stepsizes
h = 0.1 # stepsize (length of element, distance between nodes)
ha = h        # aerodynamic stepsize
hf = ha       # forces stepsize
ht=  0.000001 # stepsize for t in seconds
ht=  0.01 # stepsize for t in seconds
tend=0.2    # end time in seconds

nr_b = 1 # number of blades

# Ground effect ratio Tg/T
ge_f = 1.1

#Damping ratio
Gd= 0.03 # Calculated by NLR

# Factors. These can be set to zero to exclude or to a value between 0 and zero lower the effect on the
    response:
# 0: effect excluded, 1: effect is normal, 0-1: lowered effect, >1: amplified effect
fac_Gda = 1    # aerodynamic damping.
fac_vi = 1     # effect of vi.
fac_a = 1      # all aerodynamic effects
fac_g = 1    # gravity effects
fac_m = 1    # conservation of momentum effects
fac_cp = 1     # Effects due to centripetal forces

# Cl with vi = vi_coef * Cl without vi
vi_coef = 0.5

# Gravity
```

```
g = 9.81

rmax = 1.5-h    # Radius
r0A = 0.30     # r0, start of aerodynamic profile
r0 = 0     # r0, start of blade
osw = 0.85    # Oswald factor

# Air properties
rho = 1.225
Temp = 278 # in Kelvin
R=288  # gas constant
gamma =1.4

# for a cambell diagram
rpm_min = 0
rpm_max = 1200
rpm_step = 25
```

C.2. INPUT_AERO.CSV

An example of input_aero.csv is given below. It is a description of a simple rotor blade: it has constant span of 0.14m and a linear twist distribution ranging from 5° at the root to −3° at the tip. More rows can be inserted if required to describe the blade more accurately.

```
x;chord;Twist [deg]
0;0.14;0
0.29;0.14;0
0.3;0.14;5
1.71;0.14;-2
```

Note that the x-coordinate is slightly larger than r_{max}. This is required for the function aero_fo to create an interpolation function for the twist and chord length that can be used by the function aero up to r_{max}.

C.3. INPUT_STRUC.CSV

An example of input_struc.csv is given below.

```
x;ExAx;rhoAx;EIy;EIz;rhoIp;GIp;ynp;ycg;angle x0;angle y0;angle z0
0;28000000;10.055;70000;7000;0.001;8600;0.038;0.044;0;1.4;0
0.199;28000000;10.055;70000;7000;0.001;8600;0.038;0.044;0;1.4;0
0.2;28000000;1.055;7000;700;0.001;860;0.038;0.05;0;1.4;0
0.399;28000000;1.055;7000;700;0.001;860;0.038;0.05;0;1.4;0
0.4;28000000;1.055;14000;675;0.001;860;0.038;0.05;0;1.4;0
0.79;28000000;1.055;14000;675;0.001;855;0.038;0.05;0;1.4;0
0.8;28000000;1.055;14000;425;0.001;855;0.038;0.05;0;1.4;0
0.84;28000000;1.055;14000;425;0.001;855;0.038;0.05;0;1.4;0
0.85;28000000;1.345;14000;425;0.0015;855;0.038;0.38;0;1.4;0
1.05;28000000;1.345;14000;425;0.0015;855;0.038;0.38;0;1.4;0
1.06;28000000;1.645;14000;425;0.0022;855;0.038;0.03;0;1.4;0
1.54;28000000;1.645;14000;425;0.0022;815;0.038;0.03;0;1.4;0
1.55;28000000;1.645;10000;300;0.0022;815;0.038;0.03;0;1.4;0
1.65;28000000;1.645;10000;300;0.0022;815;0.038;0.03;0;1.4;0
1.66;28000000;1.055;10000;300;0.001;815;0.038;0.05;0;1.4;0
1.71;28000000;1.055;10000;300;0.001;810;0.038;0.05;0;1.4;0
```

Note that the x-coordinate is slightly larger than r_{max}, similar as with input_aero.csv.

C.4. INPUT_SWASH.CSV

An example of input_swash.csv is given below. This file describes a simple swash plate movement: it describes 60 seconds during which the rpm, collective and swashplate amplitude are constant at $1186 rpm$, 7.0° and 0.0°.

```
t;rpm;colective;swash ampl;v_{forw}
0;1186;7;0;0
60;1186;7;0;0
```

C.5. SET OF XFOIL DATA FILES

An example of part of one Xfoil data file out of the required set of Xfoil data files is given below. It is based on the data files created by Xfoil.exe using xfoil.py. The difference between the Xfoil.exe output file and this file is that this file as an extra column. The first column is added. It contains the x-coordinate (and thus Mach and Reynolds number) for which the calculation is done. This column needs to be added manually (using Excel or Calc is recommended).

A complete set of Xfoil data files consists of a number of files ranging from $x = 0$ to $x = r_{max}$ and each file containing a range of α similar or larger than used for calculations.

```

      XFOIL         Version 6.99

 Calculated polar for: NASA RC(4)-10 AIRFOIL

 1 1 Reynolds number fixed          Mach number fixed

 xtrf =   1.000 (top)        1.000 (bottom)
 Mach =   0.308     Re =     0.961 e 6     Ncrit =   9.000

     r      alpha      CL        CD        CDp        CM      Top_Xtr   Bot_Xtr
   -----   -------  --------  --------  --------  --------  --------  --------
   0.830   -5.000   -0.4690   0.01148   0.00526   -0.0111    0.7419    0.0232
   0.830   -4.900   -0.4574   0.01131   0.00508   -0.0108    0.7404    0.0235
   0.830   -4.800   -0.4456   0.01115   0.00491   -0.0106    0.7384    0.0239
   0.830   -4.700   -0.4338   0.01101   0.00474   -0.0104    0.7367    0.0243
   0.830   -4.600   -0.4221   0.01086   0.00457   -0.0102    0.7353    0.0247
   0.830   -4.500   -0.4103   0.01071   0.00442   -0.0100    0.7334    0.0252
   0.830   -4.400   -0.3985   0.01057   0.00427   -0.0098    0.7317    0.0257
   0.830   -4.300   -0.3867   0.01044   0.00412   -0.0096    0.7302    0.0262
   0.830   -4.200   -0.3749   0.01030   0.00397   -0.0094    0.7283    0.0269
   0.830   -4.100   -0.3630   0.01018   0.00384   -0.0092    0.7266    0.0277
   0.830   -4.000   -0.3514   0.01004   0.00370   -0.0090    0.7252    0.0285
   0.830   -3.900   -0.3395   0.00992   0.00357   -0.0088    0.7232    0.0296
   0.830   -3.800   -0.3278   0.00979   0.00344   -0.0086    0.7212    0.0309
   0.830   -3.700   -0.3161   0.00966   0.00330   -0.0084    0.7192    0.0325
   0.830   -3.600   -0.3043   0.00952   0.00317   -0.0082    0.7164    0.0350
   0.830   -3.500   -0.2928   0.00938   0.00304   -0.0080    0.7140    0.0386
   0.830   -3.400   -0.2812   0.00924   0.00292   -0.0078    0.7110    0.0437
   0.830   -3.300   -0.2695   0.00912   0.00283   -0.0076    0.7086    0.0498
   0.830   -3.200   -0.2577   0.00903   0.00275   -0.0075    0.7058    0.0553
   0.830   -3.100   -0.2457   0.00896   0.00269   -0.0073    0.7036    0.0601
```

Note that only a small part of the file is shown. Furthermore only the columns r, α, C_L, C_D and C_M are used. This means that other columns can be left out. However the order needs to remain the same. This means that although only five columns are required the aeroelastic model expects values for C_M need to be in the sixth column, unless changed in $aero\_fo$.