

AUTOMATION OF STRUCTURAL CROSS SECTIONAL ROTOR BLADE MODELLING FOR AEROMECHANICAL ROTOR BLADE OPTIMIZATION

Bram van de Kamp, Bram.vandeKamp@dlr.de,
Institute of Composite Structures and Adaptive Systems, German Aerospace Center (DLR)

Gunther Wilke, Gunther.Wilke@dlr.de,
Institute of Aerodynamics and Flow Technology, German Aerospace Center (DLR)

Abstract

It is often difficult to predict the behavior of helicopters, given their complex aeromechanical operating environments. Given these uncertainties, it is often the case that flight tests need to be conducted prior to a design being "frozen". To improve the whole design and validation procedure, it is essential that the design freeze (and iterations within) occurs before the expensive and timeconsuming process of building and flying. However, in order to do so, the virtual modelling needs to be more accurate and thus with less uncertainty. The DLR project Victoria (Virtual Aircraft Technology Integration Plattform) with its work package "Virtual Helicopter" aims to lay the foundation for a next generation comprehensive rotor code to overcome these challenges. Improvement on structural modelling within this code has high potential enhancing the overall development process, regarding time and accuracy. The improvement of rotor blade design is often driven by aerodynamic shape optimization, which means changes in airfoil shapes and sizes as well as their distribution and alignment, to fit different demands. Such changes will always have a major influence on the structural properties, because the inner structure geometry depends on the outer shape. Thus to still maintain an accurate aeromechanical model for the rotor simulation in the optimization process structural properties have to be adjusted. The most common approach is to calculate cross section data for various cross sections over the rotor blade span and feed the information into a beam-based rotor blade model. This is typically done by using approximations and scaling laws e.g., or by reducing the geometry complexity e.g. A high fidelity structural FEM-model will provide higher quality structural data. In general such FEM models are complex and require significant time to setup and process, starting with generating the blades inner geometry with CAD software, then meshing and performing the actual FEM analysis. This is very time consuming and hardly feasibly for an optimization with multiple loops. This paper presents the development of a tool for the automation of this process. The inner geometry is generated in CATIA V5 and can handle arbitrary cross section shapes (within reason). Additional parameters and boundary conditions are needed to obtain an inner geometry which is reasonable in terms of its structural integrity. These parameters include the center of gravity, basic spar shaping parameters and the skin thickness. This very accurate geometry model is then passed on to the FEM software (ANSYS). Here a mesh representing the geometry is created and then an analysis with the ANSYS tool SaMaRA is performed. SaMaRA calculates the structural properties of the cross section. The exchange of all data between the different disciplines (e.g. structural and aerodynamic) is performed via CPACS (Common Parametric Aircraft Configuration Scheme) to ensure data integrity and enable modularity of this structural code. The focus in developing this code was on the quick generation of highly accurate structural data for an aerodynamically driven optimization. The meshing automation in Ansys is not yet finished and in ongoing development. Follow up steps will be the extension of modelling options in terms of the inner structural setup and meshing quality.

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of

1. INTRODUCTION AND MOTIVATION

Helicopter rotors as main and defining part of the helicopter are not only the main propulsion source but have major influence on the helicopters behaviour in a variety of aspects. This concerns for example the fuel efficiency and the helicopter flight

this paper as part of the ERF proceedings or as individual offprints from the proceedings and for inclusion in a freely accessible web-based repository.

envelope as well as noise and vibration. These are all important topics addressed in helicopter research with impact on possible missions, operating cost and environmental influence. The rotor itself consists of various elements, major key elements are the rotor blades. They are defined by their aerodynamic shape respectively outer geometry and their structural properties. Thus one major control variable for rotor optimization is the improvement of the aerodynamic shapes of the rotor blades. The aerodynamic environment of helicopters on its own is very complex and accordingly hard to predict. But an isolated view on aerodynamics is not enough to conquer the description of helicopter behaviour. Helicopter rotor blades are elastic structures which will deform and show dynamic behaviour under the changing aerodynamic loads they are exposed to. To predict this behaviour and the interactions with the aerodynamics an accurate structural model is mandatory. Since the rotor blade structure and its properties will change with varying outer geometries, even for an optimization primarily driven by the aerodynamic rotor blade shape, the structural representation has to be updated. This structural representation most often is a beam model based on cross sectional data², data is commonly generated via scaling laws and approximations, for example like shown in¹ or with analyses on models with reduced complexity, for example seen in³. Data with increased accuracy has to be gained from cross sectional high fidelity FEA models. Since the modelling of an accurate geometry model and a dedicated FEA model needs a lot of effort and mostly multiple models over the blade radius are necessary, manually updating these models for every aerodynamic optimization loop has high temporal cost. These challenges are met in the project VicToria - Virtual Aircraft Technology Integration Platform, which is an internal DLR project. It focusses on the DLR guiding concept 6 "Virtual Product", its goal is to improve virtualization of development and manufacturing processes as well as to minimize temporal and financial cost of these processes. More specific the presented work is treated within the VicToria work package "Virtual Helicopter". Automation of the cross sectional rotor blade modelling would mean a severe acceleration of the overall optimization. Another benefit results from the fact that engineers and researchers are mostly focused on aerodynamics or structures and not both. So the aerodynamics engineer could work on the optimization, using an accurate structural representation, without going into deep detail of the structural model. Vice versa the structural engineer can focus on design principles and improving the structural model without being bound to the

aerodynamic geometry optimization directly.

2. GENERAL PROCESS OF ROTOR BLADE MODELLING

To create an automation tool it is important to know which steps have to be conducted for the model creation. In a very coarse breakdown, these are the Design step and generation of the FEA-Model. A more detailed look will be necessary here. The knowledge and experiences about the steps to be done result from earlier projects, like FTK⁶ and STAR⁴.

2.1. Design

The Design includes the cross section wise inner build up of the rotor blade and derived geometries for the FEA Model.

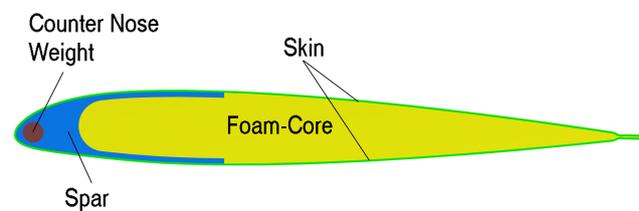


Figure 1: cross sectional inner rotor blade setup for a simple rotor blade configuration

The aerodynamic optimization will deliver an outer geometry and since the rotor blade is looked at section wise, the main input will be several profiles, these profile represent the rotor blade. Depending on the rate of change over the span the number of cross sections for a representation with sufficient accuracy varies. Looking at one specific cross section all parts of the inner structure will depend on this profile and change their size, form and/or position with it. Because the outer geometry is preset the design order from outside to inside is imperative. Looking at that for a relatively simple inner structure - meaning it only consists of a skin with uniform thickness, a C-spar, a nose counter weight and a foam core - design steps will be carried out in the following order. First the inner skin geometry has to be derived with an offset to the outer profile. This inner skin geometry is then the origin for all further geometries. Very important for the blades stability is the position of the centre of gravity being at 25%-chord length or further to the leading edge. To fulfil this boundary condition the nose counter weight is positioned as far at the leading edge as possible and its size is adjusted accordingly. For this simple configuration it will have a cir-

cular shape. Its position changes with its radius. The spar geometry fills the left over space around the nose counter weight and its main geometry will be defined by the positioning of the spar main radius and the spar flange thickness and length. The sizing of spar and weight will need some iterations. These dependencies have to be taken into account for the geometry generation part of the tool. A second part in the design process will be the preparation of the cross section geometry so that a FEA-model can be derived from it. In the later on presented case, this model will consist of two parts. This is on the one hand the rotor blade skin with a composite lay up. It is based on surface geometries, consisting of a main surface as a basis for elements and some auxiliary surfaces for element guidance. On the other hand volumes for all other rotor blade parts, with isotropic or transversal isotropic properties. Two models are to be generated for each cross section for the SaMaRA⁶ tool. Both are simple extrusions from the cross section, one with a span of 1m and the other with a span of 0,005m.

2.2. FEA-Modeling

The FEA-Modelling includes further geometry preparation for the meshing process, the meshing and the assignment of element and material properties. Geometry preparations are necessary for a shared mesh topology of multiple volumes. A mutual mesh reduces the use of contact elements and simplifies the model, although the meshing process is more complex. A structured mesh would increase the accuracy of the model but will not be a priority in the beginning, because its much harder to maintain for different geometries.

3. PRELIMINARY CONSIDERATIONS AND REQUIREMENTS

Prior to actually developing the automation tool, some thought has been put into its purpose, structure, boundary conditions and requirements. To provide high fidelity structural data, the tool must provide an accurate finite element analysis, which in turn is based on a realistic inner setup in terms of geometry and materials. Therefore the tool must cover the following tasks:

- Enable an exchange of data between the aerodynamic optimization loop, concerning both in- and output
- Read and process input data for the modelling
- Creating a detailed geometry representation

- Creating an accurate FEA model
- Integration of SaMaRA to calculate output data

Main input of the tool will be the outer geometry with direct influence on the inner setup as described in section 2.1. Further inputs are parameters, boundary conditions and material data necessary for the rotor blade description. Output will be structural data needed for the beam model. A standardized interface for data exchange eases the exchange and increases the usability. For data processing and task control a programming environment has to be chosen. In general the use of freely available software is to be preferred to reduce cost and enable use without the need of expensive licenses. Therefore the tool will be written in the programming language Python which is open source and freely available for most operating systems. Despite CATIA and ANSYS being commercial software, they are chosen for generating the Geometry and the FEA model. This choice is made because of the wide range of possibilities and options for model generation these two software products offer and also because the authors extensive prior experiences on model generation with these products. The Python environment will be used for control. In aspect of future viability the tool is created with a modular structure. The modularity allows to substitute individual parts like geometry generation or FEA modelling. This means even though commercial software is used at the moment it could be substituted by other software or own developments, if wanted. Beside the setup of the tool, the modelling capabilities of the tool are key to its area of application. Main motivation in the Victoria project is to deliver a tool for the aerodynamic optimization process. Another scenario as an area of application for the tool would for example be optimization loops for the design of active rotor blades, which are also researched at the DLR Institute for Composite Structures and Adaptive Systems, or mostly any other rotor optimization, which is not primarily driven by its structural design. But at first the focus will be on the development of a functional tool, which can generate simple, standard and at the same time realistic inner structures for a set of profiles.

3.1. Data Exchange Interface

As an exchange interface CPACS - Common Parametric Aircraft Configuration Scheme⁵ kind of puts itself forward. CPACS is a data scheme, enabling the description and characterization of aircrafts, including rotorcraft, and further aircraft related topics in XML-format. This means data is stored in an hierarchical manner and the consistency of this data

can be checked with the XSD-scheme file. Thereby CPACS is a data exchange interface with easy data access and high reliability. CPACS was created at the DLR and is in ongoing development, furthermore it is easily extendable with tool specific extensions or individual schemes. As mentioned it has a branch for rotorcraft and most of the data can be stored here. Data which is not yet accounted for in the scheme can be put into the same xml data container by extension of the XSD-scheme.

3.2. Geometry Modeling

CATIA will be used to generate the inner setup (see figure 5). As a comprehensive CAD software it has a wide range of possibilities to create 2D and 3D geometries. Since we are working with certain cross sections, the main task of generating the inner structure will be a 2D problem. Tools for automation are on the one hand scripting possibilities, in this case the COM-Interface will be used which allows to fully script our interactions with CATIA in Python. On the other hand the Powercopy function, an internal function where geometrical structures can be build on the basis of surrogate geometry objects and parameters, which can then be replaced by the actual geometries and structures. This enables a relative simple way to create complex 2D geometries. A goal should be to allow this geometry generation for a wide variety of profiles.

3.3. FEA Modelling

The FEA modelling consists of several steps, in this case all carried out within ANSYS. The first step is the meshing preparation with SpaceClaim. Second the meshing itself, which defines the grid structure and the elements properties. Also material data will be determined. And then the SamaRA-tool will calculate the blades cross section properties. Ansys offers, depending on the workbench environment, different approaches for automation. Most of them are based on IronPython, an open source implementation of the Python language. Ansys itself can be started from the command line and executed in batch mode. The project schematic can be scripted. The setup of such a script is relatively easy done by using the internal journalling function. Very similar the scripting of geometry processing in Spaceclaim has an internal journalling capability for easy scripting. The automation in the mechanical workbench of Ansys is more difficult. It is also based on IronPython and can be implemented via the Ansys Customization Toolkit (ACT). For the composite layup of the skin the Workbench Ansys Composite Pre-Post (ACP) will be used and can also be journaled

and scripted. In contrast to these automation features, needed for the rotorblade generation tool, the SaMaRA tool is based on the Ansys Parametric Design Language (APDL).

4. TOOL STRUCTURE

Figure 2 shows a flowchart of the rotor blade modeling automation tool. It consist of the CPACS data exchange interface with its sections, the Phyton control interface to process the data and operate the CAD-software CATIA and the FEA-software Ansys. The CAD and FEA modelling will not be regarded in this section, because they are already accounted for in section 2.1, 3.2 and 2.2, 3.3.

4.1. CPACS - Exchange Interface and Data Storage

As described in 3.1 the CPACS-Format has a central part in the automation tool as a data exchange and storage format. As shown in figure 2 it consists of several parts. The scheme to ensure data consistency for the exchange and then the different data containers which are only separated logically, the In- and Output Data and the CATIA_Structure. Therefore it is important to determine the exchanged data. Starting with the necessary input data the following section will give an overview on the exchanged data and modifications to the CPACS-scheme. Beside storing data, CPACS also defines standards on how to store the data and also some standards for the model itself, for example orientations of coordinate systems.

4.1.1. Input Parameters

Input parameters will take a major part in defining the inner rotor blade geometry and its properties. Most input only has to be defined initially, the only changing parameter in this case will be the profile geometry for each section, since the optimization itself is only driven by aerodynamics. Data which only has to be defined initially can be material data, basic geometric shapes and positions, geometric sizing data and additional or adjusted boundary conditions. The number and type of data differs, depending on the requirements. The basic development of the tool will stick to the simple example described in the 2.1. So the parameters will be adjusted to this case as well and can later on be used as defaults or initial values for scaling for similar profiles. The data to fill these parameters is aquired from previous projects (named in 2) to create a realistic setup.

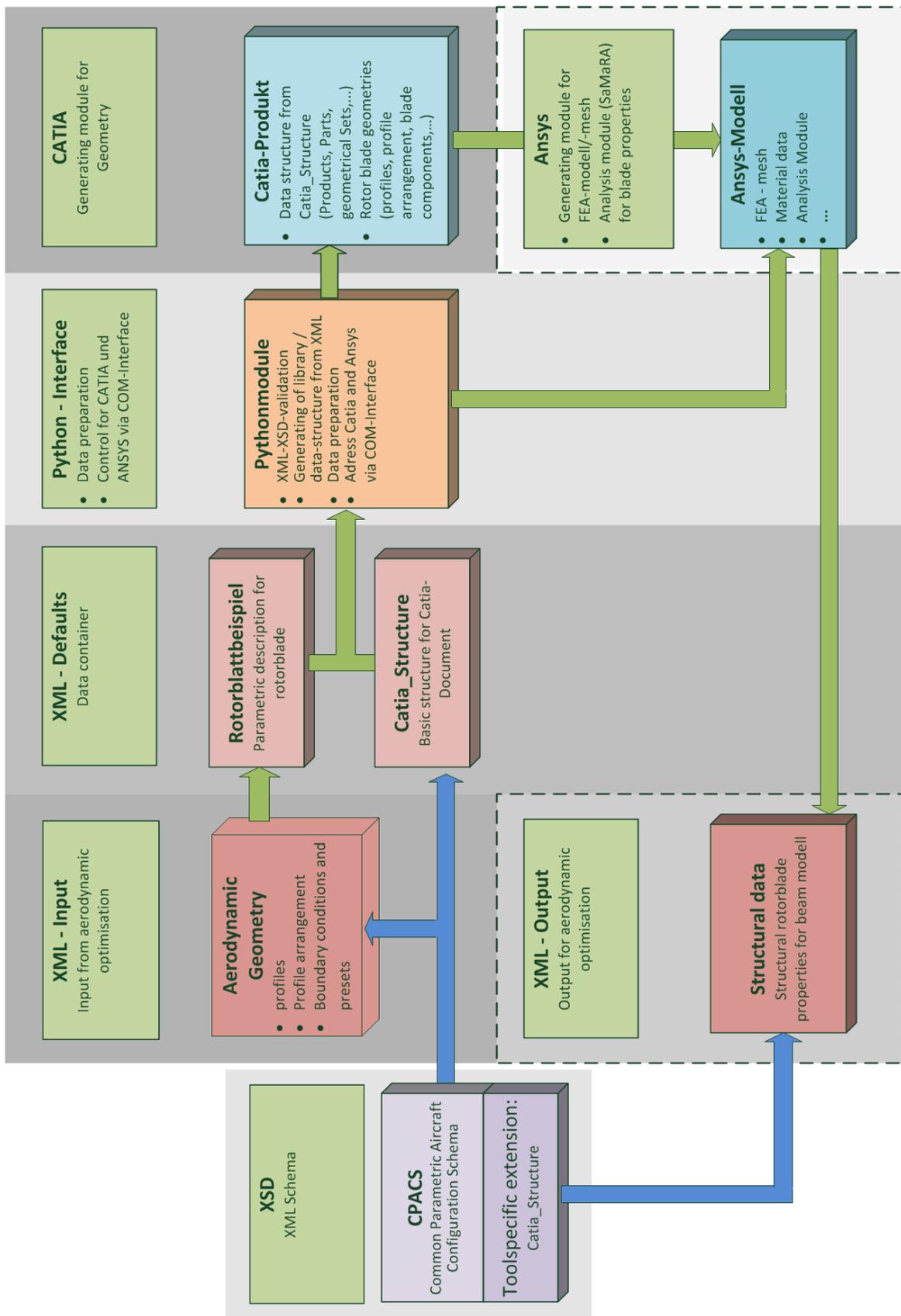


Figure 2: Tool structure

4.1.2. Output Parameters

Parameters to describe the structural behavior gained by calculation with the SaMaRA-tool are:

- Center of Gravity
- Stiffness
 - Flap bending stiffness
 - Lag bending stiffness
 - torsional stiffness
- mass per length
- mass moment of inertia
- rotor blade deformation axes
 - tension center
 - torsion center

4.1.3. CATIA_structure

For the designing in CATIA a basic structure and some basic geometrical elements are created. This step is made regarding the information stored in the CATIA_structure extension. Its is needed to enable the setup of the geometry itself. This basic structure stores the geometric elements of generated model. The basic geometric elements are for the orientation and positioning of the rotor blade according to the CPACS-standards.

4.2. Python Control Interface

The Python control interface is the core of the rotor blade modelling automation tool. It controls all tasks shown in figure 2. It reads the data from the XML input file and validates the data consistency with the extended CPACS scheme. After reading and checking the data it is processed so that it can instruct CATIA to generate the geometry model accordingly. The FEA modelling, being the following step, will also be controlled by the Python interface. The FEA model generation is still in development. It will also take place with respect to the input data.

5. EXAMPLE PROFILES BO105-SCALED AND OA209

The development of the design code used a scaled BO105⁷ profile as basis, it is shown in figure 3. In the following figure 5, the resulting inner structure is shown. A second profile with a lesser thickness and different properties is shown in figure 6, with

the resulting inner setup in figure 7. Thus the basic functionality of geometry-generation part could be shown. Further development and investigation is still ongoing, with the objective on identifying the codes limits and allow a stable geometry generation within these limits. Furthermore looking ahead pushing these boundaries will be of interest.

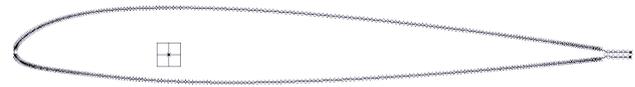


Figure 3: Point cloud for a scaled BO105 profile

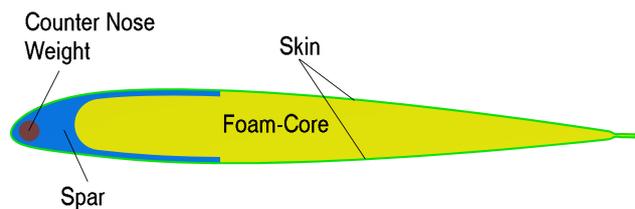


Figure 4: Cross section build up scaled BO105 profile



Figure 5: Point cloud for OA209 profile

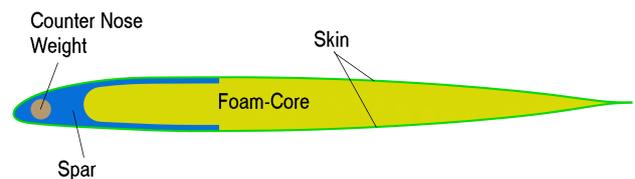


Figure 6: Cross section build up OA209 profile

6. SUMMARY AND OUTLOOK

The next steps will be to integrate the automated FEA-modelling to close the update loop and enable a fully automated structural update for changing outer profile sections. Looking into the future the geometry tool can be extended for further tasks, for example the modelling and optimization of active and morphing rotor blades. That means further geometry generation elements, like power copies have to be implemented and integrated. New parameters and options have to be added to the exchange interface. In general complexity will increase with each additional element of the inner structure and therefore the stability of the model generation will be more difficult to achieve. Probably the design space will also decrease. Another possibility would be to generate a 3D rotor blade model use this directly to calculate the structural properties. This

again would increase the complexity for the geometry generation and also for the FEA-model generation

REFERENCES

- [1] Joëlle Bailly, Biel Ortun, and Yves Delrieux. Recent advances in rotor aerodynamic optimization, including structural data update. *Journal of the American Helicopter Society*, 62(2):1–11, 2017.
- [2] Carlos E. S. Cesnik and Dewey H. Hodges. Vabs: A new concept for composite rotor blade cross-sectional modeling. *Journal of the American Helicopter Society*, 42(1):27–38, 1997.
- [3] Uttam Kumar Chakravarty. Sectionbuilder: An innovative finite element tool for analysis and design of composite rotor blade cross-sections. *Composite Structures*, 92(2):284–294, 2010.
- [4] Steffen Kalow, Bram van de Kamp, Ralf Keimer, and Johannes Riemenschneider. Experimental investigation and validation of structural properties of a new design for active twist rotor blades. In *European Rotorcraft Forum 2017*, September 2017.
- [5] B. Nagel, D. Böhnke, V. Gollnick, P. Schmollgruber, A. Rizzi, G. La Rocca, and J.J. Alonso. Communication in aircraft design: Can we establish a common language? *ICAS*, 2012.
- [6] Bram van de Kamp, Steffen Kalow, Johannes Riemenschneider, Rainer Bartels, and Henning Mainz. Multidisciplinary design chain for model rotor blades. *ICAS*, 2016.
- [7] Yung H. Yu, Chee Tung, Berend G. van der Wall, Heinz-Jürgen Pausder, Casey L. Burley, Thomas F. Brooks, Philippe Beaumier, Yves Delrieux, Edzard Mercker, and Kurt Pengel. The hart-ii test: Rotor wakes and aeroacoustics with higher-harmonic pitch control (hhc) inputs - the joint german/french/dutch/us project -. In *58th Annual Forum of the American Helicopter Society*, 2002.