

STUDIES ON ROTORCRAFT INTEGRATED DESIGN AND EVALUATION AT DLR

– FIRST RESULTS –

Max Lier^{1*}, Klaus Harbig³, Dieter Kohlgrüber³,
Alex Krenik¹, Philipp Kunze², Marius Lützenburger³

German Aerospace Center (DLR)

¹Institute of Flight Systems
Lilienthalplatz 7
D-38108 Braunschweig

²Institute of Aerodynamics and Flow Technology
Lilienthalplatz 7
D-38108 Braunschweig

³Institute of Structures and Design
Pfaffenwaldring 38-40
D-70569 Stuttgart

*Corresponding author: max.lier@dlr.de

Abstract

The automation of the rotorcraft design process is a topic of growing interest among research organisations. Another important issue in this context is the evaluation of proposed designs. To support the design process and the evaluation task, an integrated rotorcraft design and evaluation toolbox covering the conceptual and preliminary design level is developed in the DLR internal project RIDE (Rotorcraft Integrated Design and Evaluation). This paper describes the software toolbox and presents first results of the individual modules as well as lessons learned from the integration into a working process chain. The fully automated process chain comprises a basic sizing, geometry generation and structural design as well as aerodynamic and flight mechanical evaluations. The various modules have been integrated using a commercial software framework combined with a data exchange model developed in-house.

1. INTRODUCTION

The design of rotorcraft is a highly complex process. Compared to fixed-wing aircraft the design space is wider and more heterogeneous. Predesign tools are frequently used by the rotorcraft manufacturing companies. In general, they are based on detailed knowledge of existing helicopters and heavily use empirical methods. Despite this being an efficient means for the preliminary design of helicopters, the exploration of new unconventional designs is very limited. In recent years there has been a renewed interest especially in the design of unconventional concepts differing from the classical main and tail rotor configuration.

Hence, the subject of rotorcraft design has been addressed by a variety of aerospace research organisations in recent years, for example by NASA⁶, NLR⁴ or Georgia Tech⁷. The ONERA French aerospace lab started a similar project in 2011¹ in active collaboration with the German Aerospace Center (DLR), whereupon a mutual validation process has been scheduled between DLR and ONERA.

DLR is currently developing an integrated and automated tool for helicopter preliminary design and evaluation. As a collaboration of the Institute of Flight Systems, the Institute of Aerodynamics and Flow Technology and the Institute of Structures and Design the DLR internal project RIDE started in 2010 and is due to be completed by the end of 2012. The design toolbox developed provides a basis for future activities, as the basic methodology can be adopted for any rotorcraft design as long as the underlying physical models are augmented to correctly represent such configurations. Starting from a blank sheet and only using a customer specification as a driver for the whole design process it covers a wide design space and can provide high flexibility for the design solutions.

The DLR activities in the area of helicopter design are not primarily focused on the ability to design new concepts, but rather on the application of the gathered knowledge to the assessment of existing configurations. Thus the evaluation of concepts is possible even if very few data are available (e.g. if only a concept study should be assessed with respect to an intended operational use).

This paper describes the structure of the design process chain and the various tools developed in detail. First results for every design module and the current status of the integrated process chain are presented.

2. DESCRIPTION OF THE SOFTWARE TOOLBOX

DLR started to work on the development of an integrated design software toolbox in 2009. As a collaboration of the Institute of Flight Systems, the Institute of Aerodynamics and Flow Technology and the Institute of Structures and Design the DLR internal project RIDE (Rotorcraft Integrated Design and Evaluation) started in 2010 and is due to be completed at the end of 2012. The objective of the project is to provide a basis for a multidisciplinary, integrated and automated tool for preliminary rotorcraft design with a strong focus on the analysis and assessment of selected configurations.

The DLR approach divides the design process into two stages (see figure 1). Firstly, the conceptual design tool generates a suitable configuration based on the requirements thus providing a starting point for the subsequent computations. In this stage a statistical approximation is combined with simple physics-based methods and a genetic algorithm to obtain a favourable solution.

In the following stage (preliminary design) the design is refined by bringing in the perspectives of different disciplines. The geometry module uses parametric representations of helicopter fuselage geometries in a modular assembly system to generate a three-dimensional Computer-Aided Design (CAD) model (CATIA). The aerodynamic properties of the fuselage are calculated using a linearised potential code with coupled viscous boundary layer calculations (VSAERO). Then a generic structure of the fuselage is determined on the basis of static analyses of suitable load cases (ANSYS) using an automatically generated finite-element method (FEM) grid. The engine parameters are obtained by means of an exhaustive database of existing turbine engines. A statistical mass estimation utilising methods developed by Prouty¹⁵, Layton⁸, Beltramo² and Palasis¹³ completes the refined design. The resulting final configuration is at last evaluated using the flight mechanics code HOST (Helicopter Overall Simulation Tool).

The Common Parametric Aircraft Configuration Scheme (CPACS), a data exchange file format based on Extensible Markup Language (XML) developed at DLR originally for the description of fixed-wing aircraft¹¹, was augmented to incorporate rotorcraft-specific parameters and now serves as a common interface for all the software modules.

The various tools are described in detail in the following.

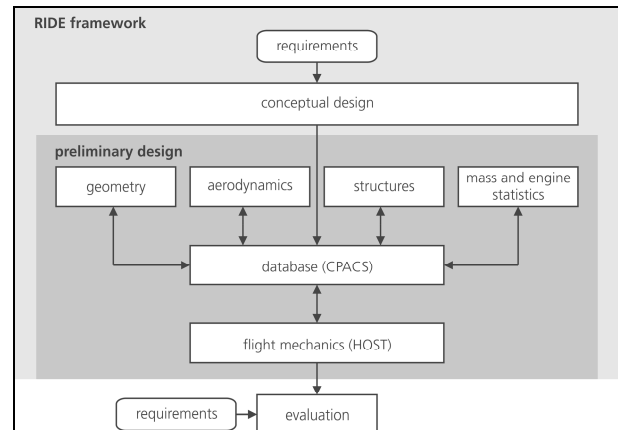


Figure 1: Process chain of the RIDE project and interactions between individual modules

2.1. Framework

The RIDE tool chain is built upon a distributed design environment developed at DLR, focusing on the application in distributed multidisciplinary preliminary design of fixed wing aircraft. The architecture and application of this design environment is described in detail by Liersch et al. in¹¹. The design environment consists of three building blocks: an integration framework, a common data exchange format plus utility and analysis modules.

The integration framework is used as a user interface for the assembly, execution and control of process chains. There are two compatible integration frameworks available at DLR: The commercial software ModelCenter by Phoenix Integration, and Remote Component Environment (RCE), an open-source framework developed in-house¹⁸. The former has been selected for use in RIDE.

The CPACS data exchange format is the key component of the DLR design environment. It is used as a common language for communication and data exchange between the user and between all integrated software modules. The development of CPACS started in 2005 within the context of the internal DLR project TIVA with the objective to create a common parametric description of the system "aircraft" suitable for all disciplines involved in the conceptual design and analysis process¹¹. XML technology has been selected for the implementation of the CPACS data exchange format. In March 2012 CPACS 2.0 has been released to the public under an open source

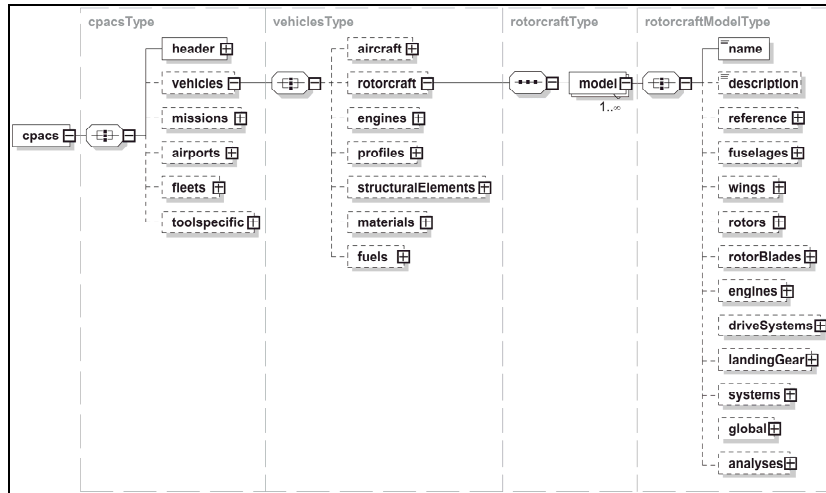


Figure 2: Root structure of the CPACS data exchange format

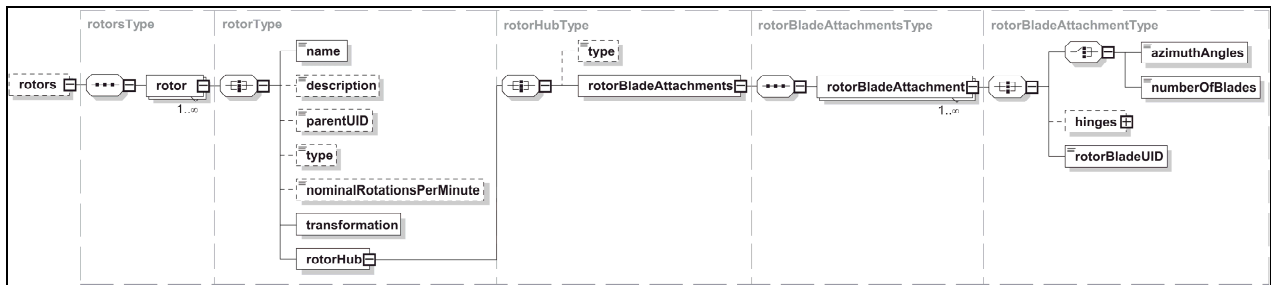


Figure 3: Structure of the CPACS rotor definition

license¹⁷. It is also gaining popularity in collaborative projects with external partners and universities^{14,16}.

For the project RIDE the current CPACS XML schema has been extended and adapted to meet the requirements of a rotorcraft design environment. Currently, most of the rotorcraft extensions reside in a development branch, but they are planned to be merged with the official CPACS version for an upcoming release. The CPACS root structure as well as a subset of child nodes is illustrated in figure 2.

The newly created `<rotorcraft>` definition is derived from the existing `<aircraft>` definition. Particularly the nodes `<rotors>` and `<driveSystems>` have been added, holding definitions of rotors and drive trains. The latter include gearboxes and their gear ratios as well as information about the interconnections between engines, gearboxes and rotors of the model. Figure 3 shows the structure of `<rotor>` nodes. Besides standard CPACS elements like `<name>`, `<description>`, `<parentUID>` and `<transformation>`, which are also present in fuselage and wing definitions, it contains information about the type of the rotor (main rotor, tail rotor, fenestron, propeller), its nominal rotation speed and, most importantly, the `<rotorHub>` element. This element

includes information about the rotor hub type and about all attached rotor blades together with the hinges used to attach them to the rotor head. In order to avoid the replication of rotor blade geometry definitions, the `<rotorBlade>` elements have been moved to a catalogue node `<rotorBlades>` of the model definition and are referenced using their unique ID inside the `<bladeAttachment>` elements. The blade geometry itself is defined using the existing data structure for the definition of wings.

Furthermore, the subnodes `<global>`, holding basic design parameters (e.g. design range and design gross weight) and `<analyses>`, storing analysis results to be shared between modules, have been adapted for rotorcraft and the tools used in the RIDE tool chain. Finally, a new child node definition has been added to the `<toolspecific>` node for each new analysis module.

2.2. Conceptual Design

The starting point of the design process chain is the COMRADE (Computer-aided Rotorcraft Assessment and DEsign) module, which performs the basic sizing of the helicopter based on the customer specification. The basic structure of the

module is shown in figure 4. The only mandatory user input is a specification or a typical mission of the helicopter to be designed. The underlying databases for the engine and mass statistics are set by default, although they can be altered by the user (e.g. to allow for restrictions of available engine types). Various calculation parameters, such as aerodynamic coefficients or parametric fuselage dimensions, are set by default. They can also be altered manually by the user in order to tailor the sizing process.

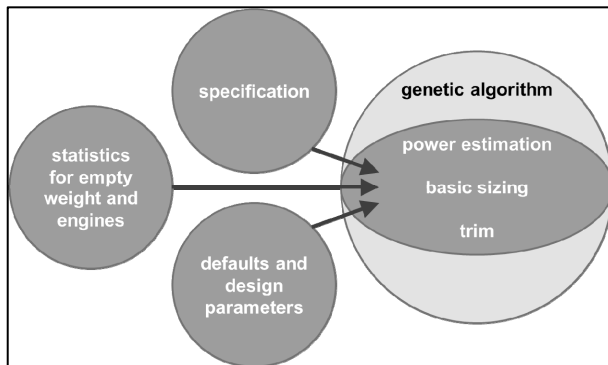


Figure 4: Basic inputs and structure of the COMRADE module

Among the parameters estimated by the module are the main dimensions of the helicopter fuselage, which are derived from the payload specification. The empty mass of the helicopter is estimated using a statistical approach based on a database of more than 140 existing helicopters. Details of the statistical method used here have been published previously⁹. Furthermore the main and tail rotors parameters are estimated using momentum theory augmented with estimations of tip loss and global separation on the rotor blades. The fuselage drag is taken into account by applying a fixed drag coefficient to the calculated fuselage drag area. After calculating the power required for the given flight condition, the engine characteristics are estimated by regression using a database of existing engines. The performance estimations are completed by calculating the required fuel weight and trimming the helicopter.

The whole sizing and performance estimation code is embedded inside a genetic algorithm used to determine a near-optimal solution for the given design specifications. The algorithm performs a heuristic search for design variables specified by the user considering constraints (i.e. minimum and maximum values), which can also be set manually along with the desired resolution of the variable. Therefore all design variables to be determined by the algorithm are combined into a string variable

(resembling a bit array) which forms the genetic representation of the design space. Afterwards a random initial population is generated. The subsequent reproduction facilitates random crossover and mutation operations and is evaluated by a fitness function which can comprise any output variables of the embedded calculation module described above.

Applied to a design specification based on the MBB Bo-105 helicopter, the algorithm provides highly satisfactory results. In this example, the genetic algorithm was initialised with a population of 100 solution candidates with the design variables to be determined being main and tail rotor diameter, the number of main rotor blades as well as main rotor solidity and tip speed. The solutions are evaluated for minimum required power (fitness function). As can be seen in figure 5, the algorithm converges rapidly over the course of twenty generations. The whole calculation is performed in less than ten seconds.

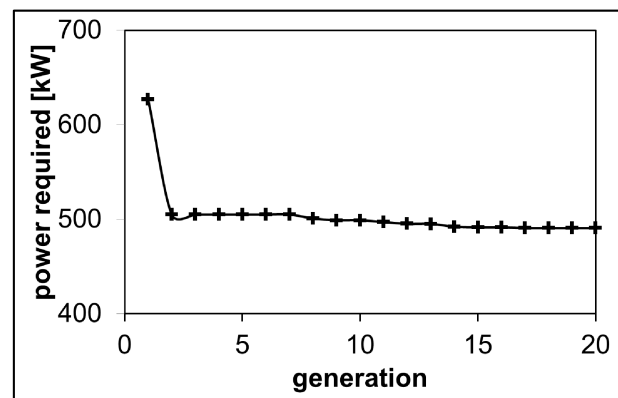


Figure 5: Convergence behaviour of the genetic algorithm for a MBB-Bo-105-like design with a population size of 100 individuals

Design studies have been done for varying forward speed and payload specifications. The results are shown in figures 6 and 7. For the forward speed variation the design solutions found by the algorithm resemble the well-known relationship between forward speed and power with the characteristic minimum required power found at medium forward speeds. Compared to the reference Bo-105 helicopter the results show a good agreement with the cruise speed and engine parameters.

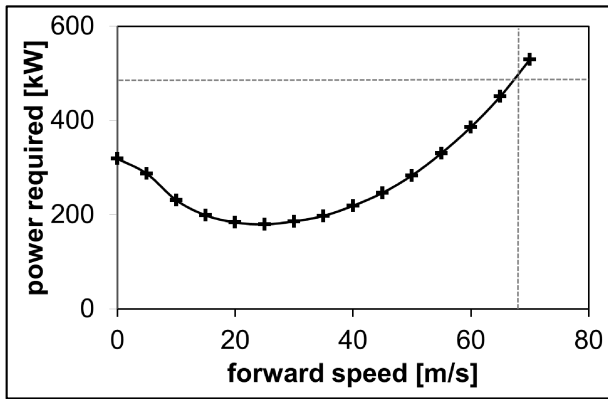


Figure 6: Design solutions for different forward speed specifications with the reference values of cruise speed and maximum continuous power of the MBB Bo-105 (dotted lines)

The payload variation leads to a significant increase in rotor radius, power required and takeoff mass for increasing payload. The results compare favourably with data of existing helicopters.

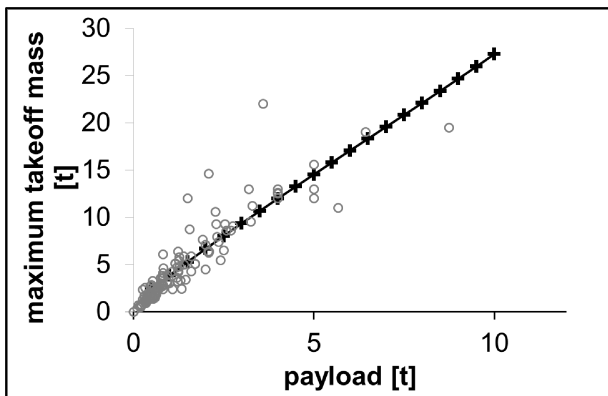


Figure 7: Design solutions for different payload specifications compared with the data of existing helicopters (grey circles)

2.3. Geometry Design

The geometry generator (GEOGEN) module automatically assembles predefined parameterised component geometry templates, adapts their parameter values and exports the resulting geometry to CPACS. In the conceptual design the design is described by a small set of global parameters (e.g. rotor diameter, fuselage length, width and height). However, higher level methods for aerodynamic and structural analyses such as panel or finite-element methods (FEM) used in the preliminary design stage require a detailed, meshable geometric representation of the model. Therefore a framework which includes the automated transition from conceptual to preliminary

design must contain an approach to create a meshable geometry model using the information available at the end of the conceptual design process, with no or minimal user interaction.

For the RIDE process chain, several possible tool architectures have been assessed and compared in terms of flexibility, expandability and development effort. As a result, a template based approach using the commercial CAD system CATIA has been selected. A collection of CATIA files containing component template definitions forms the backbone of the GEOGEN module. Component definitions include a reference to a component type, definitions of additional input parameters and geometries, and a link pointing to a CATIA template or script file used to create the geometry features associated with the component. A component catalogue containing basic components for the generation of rotorcraft fuselage geometries has been created for use in the RIDE tool chain. It comprises definitions of the following component types:

Profile2d, FuselageFront, FuselageMid, FuselageRear, Tail, EndCap, EngineCowling, Sponson, Fuselage.

Among these are e.g. three components of type Profile2d using distinct parameterisations and two components of type FuselageFront: one for helicopters with a smooth round nose (e.g. Bo-105, EC120) and one resembling the fuselage fronts of popular transport helicopters. The fuselage component can have the following geometry inputs:

Component Type	Occurrences
FuselageFront	1..1
FuselageMid	1..∞
FuselageRear	1..1
Tail	0..∞
EndCap	1..1
EngineCowling	0.. ∞
Sponson	0..∞

The script then merges all elements to build a watertight surface definition (see figure 8). For the creation of attached tail surfaces, wings and rotor blades, there is one simple trapezoidal wing template available within GEOGEN at the moment. The airfoils to be used are selected from a database and inserted to the CPACS dataset using the module AEROPole (section 2.4).

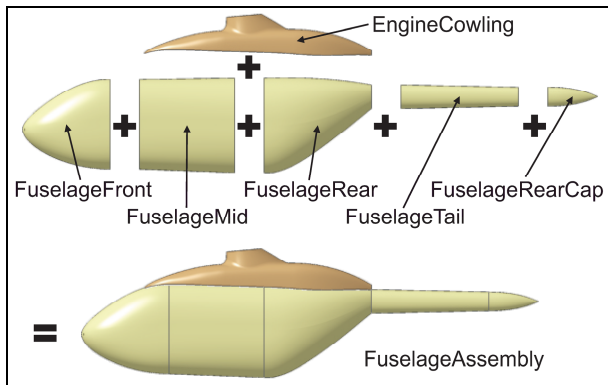


Figure 8: Simple fuselage assembly built from the GEOGEN component catalogue

The GEOGEN module has been implemented as a VBA (Visual Basic for Applications) project, making use of the CATIA Automation application programming interface (API) and the library MSXML for XML processing, input and output routines.

The program sequence can be subdivided into the following steps:

1. read assembly definition from input CPACS dataset
2. automatic creation and assembly of the components
3. set parameter values as defined in the `<toolspecific>` inputs
4. optimisation/sizing loop (optional)
5. export generated CATIA geometry to CPACS and append to the input dataset.

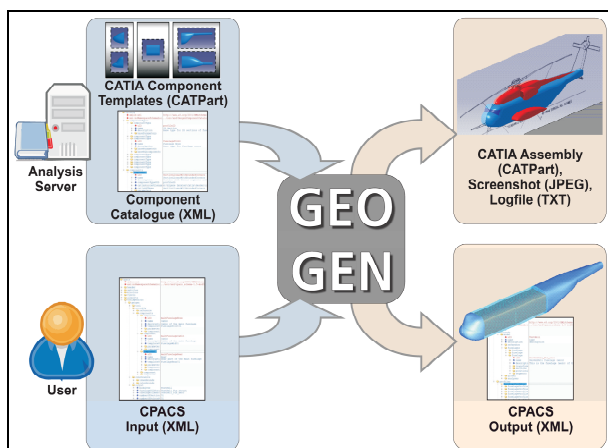


Figure 9: Inputs and outputs of GEOGEN

Figure 9 summarises the data flow and files read and written by GEOGEN. All user inputs are submitted to GEOGEN via the input CPACS file. For creation of the geometry the component catalogue

and referenced templates and scripts, residing on the analysis server, are required. A CATPart file containing the generated geometry and the output CPACS dataset including the exported geometry are returned to the user.

Finally, figure 10 shows some helicopter fuselage geometries generated using the basic rotorcraft fuselage component catalogue with GEOGEN.

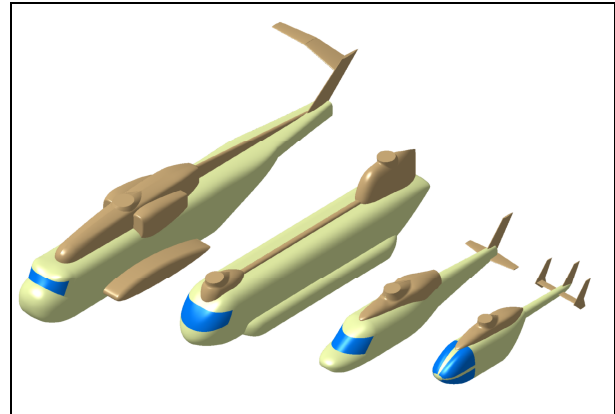


Figure 10: Examples of helicopter geometries generated using GEOGEN

2.4. Aerodynamics

The overall flight performance prediction of helicopters using the flight mechanics code HOST (section 2.7) relies on aerodynamic performance maps for isolated components (fuselages, tail surfaces, wings). For the integrated RIDE process chain these performance maps are evaluated automatically. For some components existing fixed-wing analysis modules are utilised, for others new modules had to be developed.

The module AEROFUSE generates aerodynamic performance maps of isolated fuselages. The evaluation is based on VSAERO, a linearised potential code with coupled viscous boundary layer calculations. Since the VSAERO results do not account for the viscous pressure drag caused by separated flow on the rear of blunt fuselages, a method for the estimation of the effects of this drag component on the global force and moment coefficients has been implemented.

The estimation method for viscous pressure drag is based on the assumption of constant pressure in areas of separated flow. The assumed value for the pressure coefficient in these areas can either be set by the user or determined automatically. In the latter case it is derived by calculating the mean pressure coefficient on the separation line predicted by VSAERO's boundary layer code. The effect of the viscous pressure drag on the global force

coefficients is then calculated by summing the pressure force differences due to the corrected pressure coefficient on all panels where separated flow is predicted by VSAERO ($c_f \leq 0$, grey areas of the example given in figure 11).

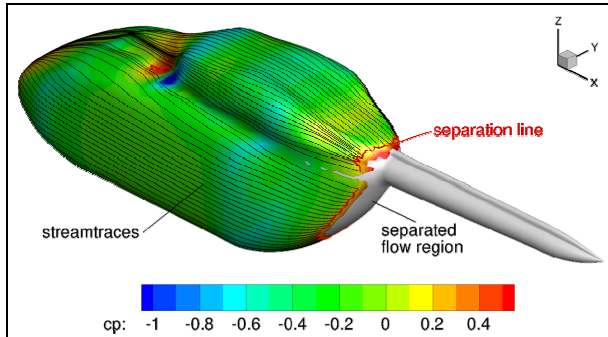


Figure 11: Surface pressure distribution, streamlines and area with separated flow predicted using VSAERO on a simplified BK117-like fuselage geometry at zero incidence and sideslip

Additional drag components, e.g. parasite drag due to attachments (such as antennas, landing gears or skids), surface imperfections or gaps can be evaluated using the existing fixed-wing analysis module HandbookAero. It enables the user to easily select predefined methods for the estimation of additional drag components (parasite, interference) using approaches found in well-known aircraft and rotorcraft design literature or based on experimental results for basic shapes, e.g.⁵.

A database containing geometry and aerodynamic characteristics of common rotor blade airfoils has been set up. The database can be accessed in the process chain using the analysis module AEROPOLE. It simply copies the data for selected airfoils from the database to the CPACS dataset. The functionality has been encapsulated into one tool to ensure consistency of the data in the CPACS dataset. Furthermore it prepares the process chain for the projected automatic evaluation of airfoil aerodynamic characteristics.

For the evaluation of aerodynamic characteristics of tail surfaces and wings existing analysis modules of various fidelity levels developed for fixed-wing preliminary design can be invoked:

- The module *HandbookAero* provides predefined handbook methods or user defined functions to create or modify aerodynamic performance maps.
- The *LiftingLine* module is a wrapper around the DLR in-house multi-lifting-line code LIFTING_LINE¹⁰.

- The *VsAero* module is a wrapper for the evaluation of wing aerodynamics using the proprietary coupled panel and viscous boundary layer code VSAERO.

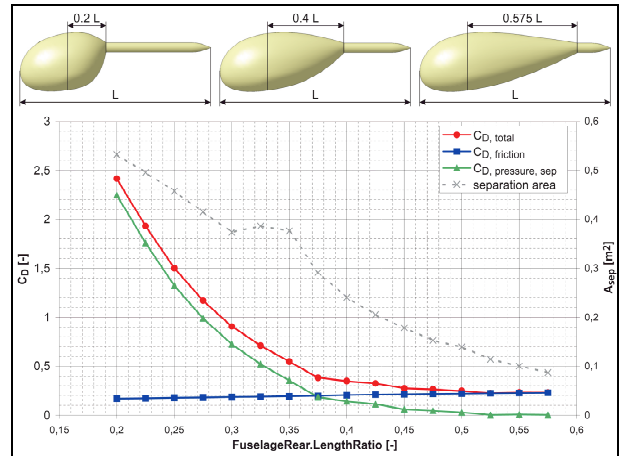


Figure 12: Parameter study coupling GEOGEN and AEROFUSE

Figure 12 shows the results of a parameter study coupling GEOGEN with AEROFUSE. The fuselage drag at zero incidence and zero sideslip is evaluated as a function of the fuselage rear segment length ratio. The drag calculated by AEROFUSE seems plausible, but further investigations for the calibration and validation of the method have to be carried out.

2.5. Improved mass statistics

The RIDE module MASERATI (Mass Estimation of Rotorcraft based on Statistics) calculates the rotorcraft masses based on equations which were developed by the evaluation of existing aircraft. Here the level of detail is low and currently limited to the weight of the complete respective group (e.g. body group, landing gear group or drive system).

Presently, 4 different methods for the calculation of the weights are implemented in MASERATI: Prouty¹⁵, Layton⁸, Beltramo² and Palasis¹³. The respective statistical equations were taken from literature and are based on historical trends. In some cases, single equations had to be corrected since there were obviously mistakes in the printed source which was on hand. In order to avoid potential new mistakes, all equations were implemented in the original imperial units. The user can select which units are used for the input and output data. MASERATI requires up to 56 different input variables. If all these variables are accessible, the program can calculate the weights with all the currently implemented methods and compare the different results. In many cases, not the complete

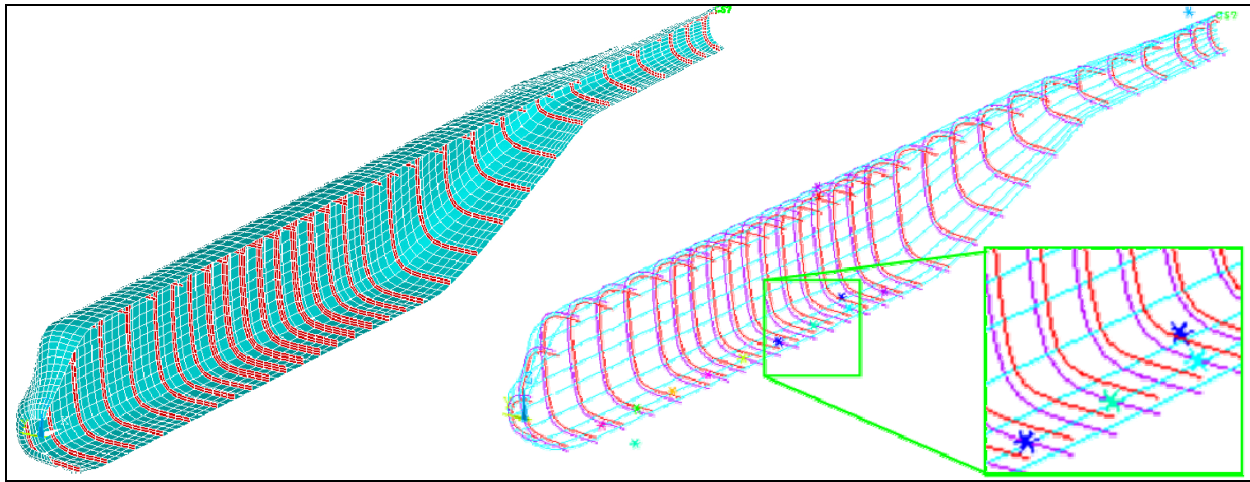


Figure 13: ROFUMA half model of a generic helicopter fuselage

set of variables will be defined. Then, MASERATI selects the calculation method which fits the best with the available variables. In MASERATI, only the weights of the single systems/groups are currently calculated; the respective mass positions as well as their moments of inertia cannot be calculated. The mass positions are defined within the RIDE module GEOGEN and transferred to MASERATI where the overall moments of inertia for the complete rotorcraft are calculated by use of the Steiner theorem. The structure of the MASERATI output data is based on the document "SAWE RP No. 8A – Group Weight Statement AIRCRAFT (Including Rotorcraft)" which was issued by the Society of Allied Weight Engineers (SAWE).

For the future, it is planned to introduce some factors into the calculation process which take into account the technological progress achieved with new materials and design methods. Due to limited resources and the lack of complete datasets for newer aircraft this could not yet be realised. Up to now MASERATI has only been validated by use of the data of "historical" rotorcraft. New configurations have not yet been tested.

For the estimation of the rotor blade mass and moment of inertia a generic rotor blade structure (C-spar or D-spar design) is scaled with respect to the airfoil definition and blade dimensions. By integration along the blade length the blade mass and centre of gravity as well as the first and second moment of area are determined.

2.6. Structural Design

The development of the RIDE module ROFUMA (ROtorcraft FUselage Mass Assessment) has been initiated to enable a physics-based estimation of the mass of the rotorcraft fuselage structure. This approach aims to allow a more detailed assessment

of the rotorcraft structural mass (body group) compared to the statistical approach in the MASERATI module and finally the direct comparison of different design variants, such as number of frames or material selection (e.g. metallic vs. composite).

The ROFUMA module itself consists of three parts which are run subsequently. In a first step a Finite Element (FE) model is generated from the structural description in the corresponding CPACS data file. After the mesh generation the non-structural masses as well as external loads such as rotor loads or aerodynamic drag are applied to the model according to a list of relevant load cases. In a second step static FE simulations of the considered load cases are performed using the ANSYS FE solver. In a final step yet to be implemented the sizing of the finite elements (thickness for shell elements, cross sectional area for bar and beams) based on strength and simplified stability criteria will be performed in the so-called S-BOT (Sizing-Robot) environment initially developed at DLR for the sizing of transport aircraft wings¹².

Currently the first two steps of the ROFUMA module have been implemented with some limitations in the mesh representation, which will be explained in more detail in the following. The FE mesh generator is programmed partly in Python and the specific programming environment of the ANSYS PRE-Processor PRE7 (APDL: Ansys Parametric Design Language). This mesh generation is fully based on data which are defined in a CPACS data file. Currently the following structural components are modelled:

- **Skin:** The skin is modelled using shell elements. The properties (material selection and sheet thickness) are defined according to different panel definitions in the CPACS file. A

panel definition for composite materials is not supported in the current version.

- **Stringers:** The stringers (more generally: longitudinal reinforcements of the skin) are modelled using beam elements along the fuselage skin. The properties of the beam elements are calculated automatically from the corresponding definitions of the structural elements in CPACS.
- **Frames:** The frames are modelled using beam elements or a combination of shell and beam elements, which are modelled at the positions defined in the CPACS file. The properties of the beam elements are calculated automatically from the corresponding definition in CPACS.
- **Floor structure, doors, windows:** Neither a floor structure nor cut-outs in the fuselage shell, such as passenger or cargo doors and windows, are modelled until now. The additional mass of these structural components may be estimated and considered as additional masses in the global structural analysis. However, a more detailed representation of these cut-outs will be added in future versions of ROFUMA.

In figure 13 exemplary pictures of a ROFUMA mesh generated on the basis of a generic heavy transport helicopter surface and a generic distribution of internal reinforcements are presented. While the shell elements to discretise the outer skin and the frame webs are plotted in the left hand picture, the beam elements that are used for stringers (light blue) and for the inner and outer frame caps (red and violet) are presented on the right hand side. In addition to the beam elements, some mass point elements to represent the additional masses added to the model are shown in this plot (coloured asterisks).

In figure 14 the beam elements of the model shown in figure 13 are displayed using the ANSYS /ESHAPE option that plots beam elements with their real cross sectional area together with the frame webs in shell representation. This cross section is automatically calculated from the CPACS description of the structural elements. In this example the stringers are modelled as hat profiles with increased height in the floor region, and rectangular solid sections are defined for the frame caps. However, the shape of the profiles is completely arbitrary and can be defined in the CPACS file using a profile definition via point coordinates and the sheet thickness between the profile points.

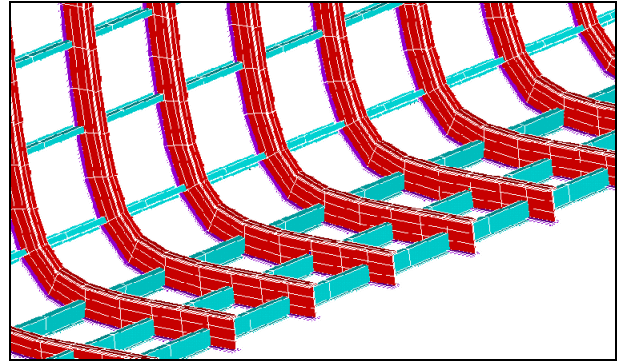


Figure 14: Stringers and frames with display of the real cross sections

In addition to the mesh generation described above, added masses as well as external loads are modelled. In each load case definition, a variable number of external loads can be defined (e.g. for the main and tail rotor), each with 3 components for forces and moments, combined with the size of a rigid region, which is used to introduce the load into the structure in a simplified way. In the FE solver all nodes in this rigid region are coupled using constraint equations. Although this approach prohibits a detailed strength analysis and mass assessment in the load introduction area, an adequate transfer into the fuselage structure is provided. In addition to the external loads a constant acceleration field acting on the entire fuselage can be defined.

The non-structural masses in the fuselage (incl. systems, seats, payload, etc.) are then added to the model. Each static mass definition consists of a scalar value of the mass and a centre of gravity in global Cartesian coordinates. This static mass is distributed over a certain group of nodes using interpolation rigid body element (RBE3) constraints. The selection of nodes to which the added mass is distributed is defined for each additional mass in a mass influence region. In Figure 15 an exemplary distribution of three additional masses on the floor to the inner frame caps is illustrated.

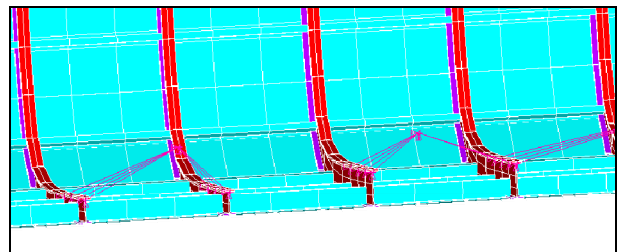
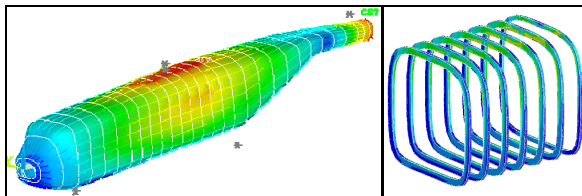


Figure 15: Illustration of the distribution of 3 additional masses to the inner flanges of the frames via RBE3 constraints

After the model generation is completed, static analyses of selected load cases are started by the ROFUMA module automatically. As a result of these analyses the deformations as well as the stress levels in the structure are calculated. In Figure 17 some exemplary results from a generic ROFUMA simulation run of a +2 g manoeuvre load case are presented in the plot on the left hand side. The high loads in the helicopter main frames below the high rotor, engine and main gearbox masses are highlighted in the plot on the right hand side of Figure 17, which displays von Mises stresses in these frames.



As described above, the ROFUMA module includes a complex internal process chain with FE model generation and a subsequent iterative structural sizing process in an FE environment based on an

2.7. Flight Mechanics Evaluation

HOST (Helicopter Overall Simulation Tool) is a FORTRAN 77 based software developed in 1994 by the aerodynamic department of Eurocopter France (EC). HOST is an integration structure for all the existing and future helicopter simulation tools modelling the mechanics of the flight of the helicopter and its dynamic behaviour in flight or on the ground³. Since its introduction HOST was continuously extended with new models and new functionalities (FCS, dynamic engine models, additional rotor induced velocity models, etc.).

The user has two options of executing the calculations: either interactively via the graphical user interface or utilising the batch mode with pre-set conditions, boundaries etc. In the ModelCenter environment only the non-interactive method is relevant. The dataset structure of HOST is modular with the main configuration file to link all the functionalities and dependencies among the mostly physics-based helicopter modules (figure 18).

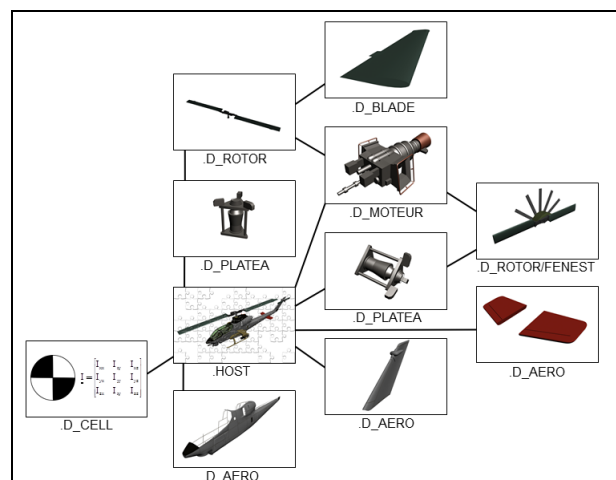


Figure 18: Setup of HOST input files for a dataset representing a conventional helicopter configuration

The depth of the modelling is also variable (actuator disc, blade element method, rigid or soft blades). It is necessary for the complete integration to transfer the data from CPACS records into HOST input files, to enable an automated HOST execution and to process the obtained data in the design environment of ModelCenter. The automated generation of a complete HOST input data set based on the CPACS input is performed by supplementary routines written in Python programming language.

Among the multitude of the design and configuration parameters in the preliminary design of the helicopter there is the non-negligible impact of the blade number on the total required power. Figure 19 shows the variation main rotor rotational speed and blade number on the total power in the steady horizontal speed of the Bo-105 helicopter at 100 km/h airspeed. Thanks to the integration into the ModelCenter framework without the necessity of any user interactions, those trade-off studies can be performed very quickly.

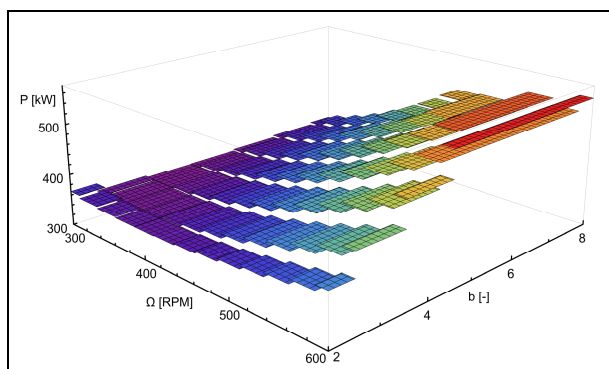


Figure 19: Results for the total required power (P) for a variation of main rotor rotational speed (Ω) and blade number (b)

3. INTEGRATION RESULTS

The use of CPACS as a common interface for all modules has proven to be successful. Thus, the definition of individual interfaces between interacting modules is not necessary, ensuring maximum flexibility regarding the process chain. Additionally, intermediate results are easily accessible and can be examined as all data generated by any module at a certain stage of the process are present in the CPACS data set exchanged between the tools.

The whole process chain only takes the customer specification as a mandatory input. Currently the specification variables are cruise speed and range as well as payload and cabin volume. Additionally a variety of optional variables can be passed to the individual modules in order to directly influence the underlying models of the various calculations and

therefore to adjust the process chain to the specific design problem.

The COMRADE tool provides a basic helicopter model, which can then be further refined by the subsequent modules. Experience has been that the automated geometry generation works for any plausibly sized solution generated by COMRADE. The ROFUMA module is fully integrated into the RIDE process. All data required for the model generation and the subsequent static analyses of considered load cases are extracted from the CPACS data file through a wrapper tool. Most of the input data are output from the COMRADE, GEOGEN and MASERATI modules. Other parameters, such as the individual definition of all structural reinforcements (stringers, frames) are automatically generated by ROFUMA in a very first step. The use of HOST inside the RIDE process chain is possible without user interaction. All necessary input data are generated within the process chain, allowing for a complete description of the helicopter in HOST input format. The evaluation of the design solution can then be performed using the extensive data calculated by HOST.

The process chain has been implemented in ModelCenter and is fully functional. One single calculation of the whole chain is executed in well under five minutes, which should be sufficient for a preliminary design study.

4. CONCLUSION AND OUTLOOK

In the course of the DLR project RIDE a truly integrated design environment for transport helicopters has been developed. A three-dimensional geometry and detailed structural design solution accompanied by profound data on flight performance calculated by HOST is determined starting from very few customer specifications. The RIDE toolbox will further be extended to allow the design of unconventional configurations, where a simplified modelling of tandem rotor arrangements is scheduled for the end of 2012.

There are a number of improvements which are planned to be added to the individual modules in the future. The conceptual design model could be further improved by adding empirical estimations for design parameters currently represented by fixed default values. The modelling of unconventional configurations is another field of work for the sizing module. Concerning the geometry generator, more complex fuselage components could be added in order to allow for a more flexible geometry design. In addition to the final implementation of the structural sizing, the parametrical model generation should be extended to consider further structural elements such as the floor structure as well as fuselage cut-

outs. As the engine is a crucial part of every helicopter design, the engine database should be substituted with a parametric, fully scalable and physics-based engine model.

Regarding the whole process chain an iterative optimisation procedure has to be introduced as currently the only sizing task is done using the relatively simple model in the conceptual design phase. However, in order to further improve the design result all modules have to be integrated in a global sizing loop.

5. REFERENCES

- [1] Basset, P.-M., Tremolet, A., Cuzieux, C., Schulte, D., Trisant, D., Lefebvre, T., Reboul, G., Costes, M., Richez, F., Burguburu, S., Petot, D., Paluch, B.: The C.R.E.A.T.I.O.N. Project for Rotorcraft Concepts Evaluation: The First Steps, 37th European Rotorcraft Forum, 2011.
- [2] Beltramo, M. N., Morris, M. A.: Parametric Study of Aircraft Systems Costs and Weights, National Aeronautics and Space Administration, NASA CR-152315, 1980.
- [3] Benoit, B., Dequin, A.-M., Kampa, K., von Grünhagen, W., Basset, P.-M., Gimonet, B.: HOST: A general helicopter simulation tool for Germany and France, American Helicopter Society, 56th Annual Forum, 2000.
- [4] Boer, J.-F., Stevens, J.: Helicopter Life Cycle Cost Reduction through Pre-Design Optimisation, 32nd European Rotorcraft Forum, 2006.
- [5] Hoerner, S.: Fluid Dynamic Drag: practical information on aerodynamic drag and hydrodynamic resistance, Hoerner Fluid Dynamics, Midland Park, 1965.
- [6] Johnson, W.: NDARC - NASA Design and Analysis of Rotorcraft Validation and Demonstration, AHS Aeromechanics Specialists' Conference, 2010.
- [7] Khalid, A. S., Schrage, D. P.: Helicopter Design Cost Minimization using Multidisciplinary Design Optimization, 63rd Annual Forum of the American Helicopter Society, 2007.
- [8] Layton, D. M.: Helicopter Conceptual Design, 1991/1992 AIAA Home Study Correspondence Course.
- [9] Lier, M.: Statistical Methods for Helicopter Preliminary Design and Sizing, 37th European Rotorcraft Forum, 2011.
- [10] Liersch, C., Wunderlich, T.: A Fast Aerodynamic Tool for Preliminary Aircraft Design, AIAA-2008-5901, 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2008.
- [11] Liersch, C. M., Hepperle, M.: A Distributed Toolbox for Multidisciplinary Preliminary Aircraft Design, CEAS Aeronautical Journal (2011) 2:57-68.
- [12] Nagel, B., Kintscher, M., Streit, T.: Active and Passive Structural Measures For Aeroelastic Winglet Design, 26th International Congress Of The Aeronautical Sciences, 2008.
- [13] Palasis, D.: Erstellung eines Vorentwurfsverfahrens für Hubschrauber mit einer Erweiterung für das Kipprotorflugzeug, PhD thesis, VDI Fortschrittsberichte 7 / 201, 1992.
- [14] Pfeiffer, T., Nagel, B., Böhnke, D., Rizzi, A., Voskuil, M.: Implementation of a Heterogeneous, Variable-Fidelity Framework for Flight Mechanics Analysis in Preliminary Aircraft Design, DGLR Congress, 2011.
- [15] Prouty, W. R.: Helicopter Performance, Stability, and Control, Krieger Publishing Company, Malabar, Florida, 1995.
- [16] Rizzi, A., Zhang, M., Nagel, B., Böhnke, D., Saquet, P.: Towards a Unified Framework using CPACS for Geometry Management in Aircraft Design, AIAA-2012-549, 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2012.
- [17] Böhnke, D.: CPACS – A Common Language for Aircraft Design, <http://software.dlr.de/p/cpacs/home/>, accessed: June 2012.
- [18] Seider, D., Bachmann, A., Kunde, M., Litz, M.: Remote Component Environment, <http://software.dlr.de/p/rcenvironment/home/>, accessed: June 2012.