

PARAMETRIC FUSELAGE GEOMETRY GENERATION AND AERODYNAMIC PERFORMANCE PREDICTION IN PRELIMINARY ROTORCRAFT DESIGN

Philipp Kunze, Philipp.Kunze@dlr.de, German Aerospace Center (DLR)

Abstract

The creation of an integrated rotorcraft conceptual and preliminary design framework at DLR involved the development of geometry and fuselage aerodynamics modules at the Institute of Aerodynamics and Flow Technology. After a short revision of the RIDE rotorcraft design environment architecture this paper focuses on the implementation of these disciplinary modules. The aim of the geometry module is to bridge the gap between conceptual and preliminary design and to allow for geometry parameter and optimization studies in the preliminary design phase. An overview of geometry generation concepts is given. Then the selected and implemented template-based geometry generation process based on the commercial CAD system CATIA and its integration into the design framework are described. Furthermore the underlying component catalog concept is illustrated considering the example of basic templates for the generation of rotorcraft geometry components. The fuselage aerodynamics module presented in this paper is based on the commercial panel code VSAERO combined with a simple estimation for pressure drag due to separated flow. The aerodynamics module is applied to common testcases and compared to experimental results. Finally, the usage of the newly developed modules within the preliminary design framework is demonstrated by coupling them in a toolchain, showing the effect of geometric parameters on the predicted aerodynamic performance of the fuselage.

1. INTRODUCTION

Classic aircraft and rotorcraft design textbooks^{[21],[2]} usually divide the design process into three distinct stages: conceptual design, preliminary design and detail design.

The conceptual design stage follows the decision for the development a new rotorcraft. A specification of requirements imposed on the new product on the basis of customer requirements or internal strategic directions is used as input for the conceptual design. With that, designers try to find the vehicle concept, which best meets all these requirements. This task requires a holistic approach in order to find optimal configurations in a huge multidisciplinary design space, typically involving considerations in aerodynamics, structures, weight and balance, flight dynamics performance and costs, but also regarding constraints, e.g. set by airworthiness standards. It is usually accomplished by first narrowing the design space using knowledge and experience of the design engineers along with market analyses and statistics and selecting a few promising rotorcraft concepts to be further evaluated. Then analysis methods based on statistics, empirical formulas or simple physical models yield to a quantitative comparison of the designs and assist the designer in setting initial values for basic parameters which describe the finally selected rotorcraft configuration. The final product of conceptual design is a layout of this configuration showing the basic arrangement and location of all relevant components (e.g. fuselage, rotors, tail, tail surfaces, engines) and their most basic dimensions (e.g. fuselage length, rotor diameter...).

In the second stage of the design process, the preliminary design, the configuration from the conceptual design stage is further refined and optimized (e.g. definition of fuselage surface, rotor blade number and planform, stabilizer sizes and planforms, basic structural and control features, noise characteristics). The disciplinary methods applied reach from textbook methods to sophisticated numerical meth-

ods (e.g. comprehensive rotor codes, CFD, FEM). At the end of the preliminary design phase the configuration is frozen.

Finally, in the detail design phase the rotorcraft design is turned into a complete manufacturable product definition. It involves the definition and dimensioning of all structural elements, control systems, the drive train, electronics and avionics.

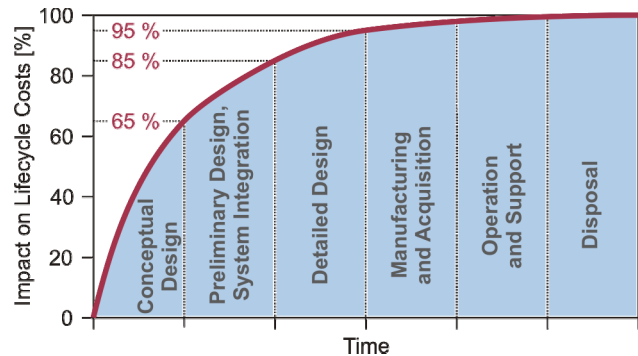


Figure 1: Lifecycle costs according to Roskam^[18]

The success of a new rotorcraft is significantly influenced by the decisions made during its conceptual and preliminary design phases. Figure 1 shows that approx. 85% of the lifecycle costs of an aircraft are determined in these early design stages. Moreover mistakes made in conceptual or preliminary design can remain undetected until late detail design, manufacturing or even until the testing phase causing considerable delays and economic burden in the product development. Therefore it is desirable to improve the quality of the methods used in conceptual and preliminary design in order to minimize uncertainties and to reduce the risk of making mistakes in these design stages. This can be achieved with a higher level of automation, streamlining the workflow and reducing uncertainties by integrating higher fidelity physics based methods earlier in the design process. Thus, for example, a high fidelity aerodynamic analysis during preliminary design can

be used to detect stability issues otherwise possibly undetected until flight testing and avoid costly design changes late in the development process. The availability of productivity enhancing software and the continuous progress in computer hardware performance facilitate the implementation of such measures.

Within the project RIDE (Rotorcraft Integrated Design and Evaluation)^[11] the DLR started the development of an integrated rotorcraft design environment with focus on seamless streamlined processes starting from conceptual design till physics based disciplinary analyses in the preliminary design context. The aims of this project were to establish knowledge and competence in rotorcraft design at DLR and also the ability to quickly evaluate new or unconventional rotorcraft technologies and configurations. A modular data-centric architecture used for fixed wing design activities at DLR has been adopted for the new rotorcraft design environment. The extension of the data format and the implementation and integration of a geometry module and aerodynamic analysis modules into the design environment were the main tasks accomplished by the Institute of Aerodynamics and Flow Technology in RIDE. This paper focuses on these activities, starting with a revision of the general architecture of the RIDE rotorcraft design environment. Then the development and integration of the geometry generation module GEOGEN and the fuselage aerodynamics module AEROFUSE into the RIDE design framework are discussed. The analysis results of AEROFUSE are compared to available experimental data for selected testcases. Finally an application example illustrates the use and functionality of the presented tools within the RIDE design environment.

2. RIDE ROTORCRAFT DESIGN ENVIRONMENT

Figure 2 outlines the basic concept of the RIDE toolchain. It consists of a conceptual design module followed by a preliminary design loop with disciplinary analysis modules. All modules are coupled using a central data exchange format (CPACS). The range covered by the conceptual and preliminary design modules is geared towards performance centric rotorcraft design, as factors including costs or noise are not considered in the toolchain yet.

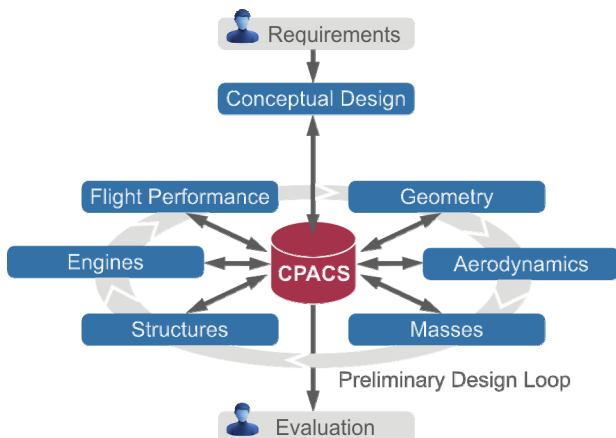


Figure 2: RIDE toolchain

The institutes involved in the development and application of the RIDE modules are physically distributed across several DLR sites, requiring a distributed architecture of the design environment. A major aspiration in the planning

and the setup of the RIDE project was the transfer of knowledge, experience and software, which have been aggregated at DLR in similar fixed-wing projects in the years before RIDE has been started. Consequently the architecture of the available data-centric collaborative aircraft design environment depicted in Figure 3 has been chosen as a basis for the RIDE rotorcraft design environment.

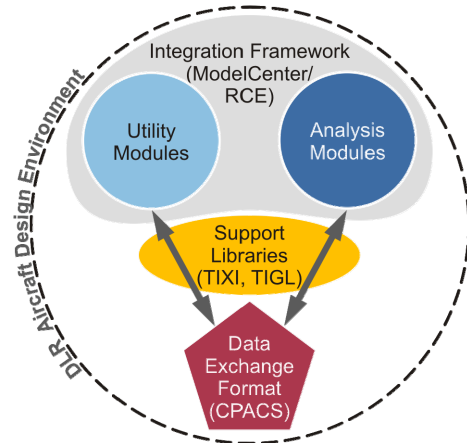


Figure 3: Building blocks of the DLR aircraft design environment

The design environment consists of three main elements: a common *data exchange format*, an *integration framework*, and *analysis modules*, usually linked by utility libraries and utility modules. The following sections detail the concept and intention of these building blocks and adaptations made in the RIDE project for their application to rotorcraft design. These adaptations include the extension of the data format definition and the development of new analysis modules, while only minor adaptations in the utility libraries were required and the integration framework could remain completely unchanged.

2.1. Data Exchange Format

The main element of a data-centric architecture is a data format definition used as a common language for data exchange among all utility and analysis modules, and also the user. The specification of a data format is a major effort that pays off with an increasing number of analysis modules in the design framework.

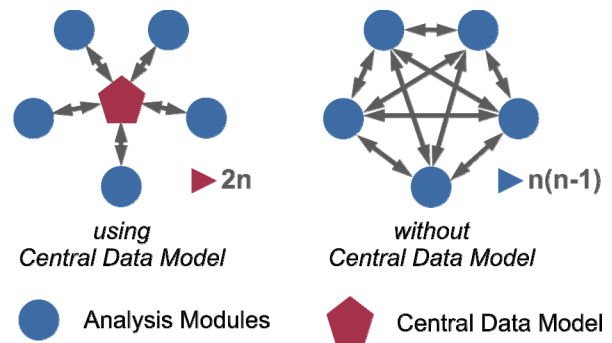


Figure 4: Number of interfaces with and without central data model according to^[17]

Figure 4 visualizes one major advantage of such an approach compared to an approach without central data model: It minimizes the number of required interfaces, while retaining maximum flexibility in the setup and rear-

rangement of toolchains. For example, a disciplinary analysis module can easily be replaced by a module with similar functionality and a different level of fidelity or moved to another position in the sequence of analysis modules without reimplementing any interfaces. Furthermore standardized interfaces ease the reuse of developed modules in other tasks or projects and thus potentially reduce future development efforts.

CPACS (Common Parametric Aircraft Configuration Scheme) is the data exchange format developed at the DLR for its multidisciplinary aircraft design activities starting from 2005. Since then the format specification is being extended and revised on an ongoing basis, covering an ever growing number of disciplines and application fields. In 2012 CPACS has been released to the public as open-source^[6] and has subsequently also been gaining popularity outside the DLR, in research institutes, universities and industry. Consequently CPACS has been selected as core technology for the rotorcraft design toolbox developed. In the early RIDE project phase several extensions have been defined in order to address rotorcraft specific tasks. They were compiled into a RIDE-internal CPACS version (CPACS4RIDE). These extensions will be reviewed and likely be included in an upcoming official CPACS release.

CPACS is based on XML technologies. The data format is specified using an XML schema description (XSD) with embedded documentation. It uses a hierarchical data structure storing both product (e.g. global aircraft properties, geometry definitions, analysis results) and process information (e.g. toolspecific inputs for all analysis modules in the toolchain). Figure 5 depicts essential elements of the CPACS structure and marks the element definitions which have been created or modified in the version CPACS4RIDE.

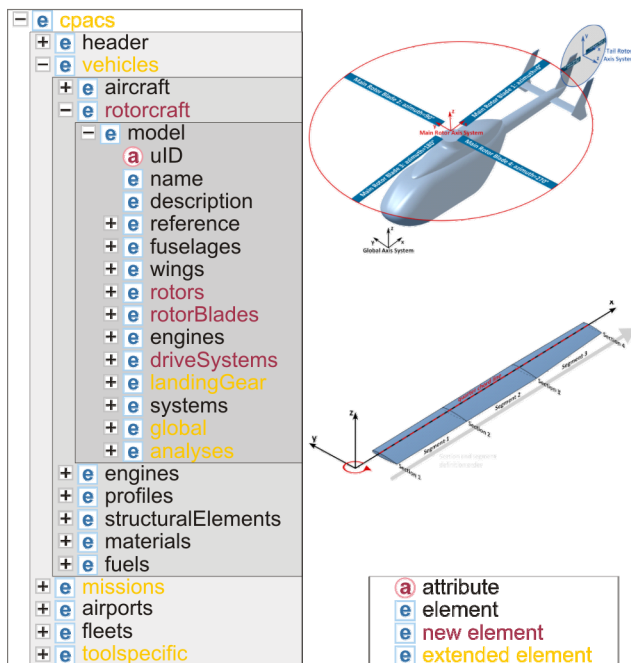


Figure 5: CPACS4RIDE root structure

The definition of geometrical components (fuselages, wings, rotor blades...) in CPACS is based on a generic parameterization optimized for the representation of common wing geometries (see^{[12], [6]}).

2.2. Integration Framework

The integration environment performs two main tasks. On the one hand it provides a user interface for the design environment. Its GUI lets the user create, setup and control toolchains. On the other hand the integration framework handles the data exchange, access privileges and flow control behind the scenes in a distributed environment, consisting of analysis modules and clients running on multiple computers spread across the network. Finally the integration framework also provides tool developers with an infrastructure for the integration of analysis tools.

The commercial software ModelCenter^[15] by Phoenix Integration has been selected as integration framework for the project RIDE. DLR has developed plug-ins and extensions for ModelCenter which ease the handling of CPACS datasets in the ModelCenter client and the development of CPACS-based analysis modules for the AnalysisServer.

2.3. Analysis Modules

While the building blocks discussed so far provide the infrastructure for the design framework, its core functionality is given by the integrated analysis modules. In RIDE, an analysis module for conceptual design, and disciplinary analysis modules covering geometry generation, weight estimation, aerodynamics and flight dynamics, have been developed. Besides, some existing analysis modules from the fixed-wing aircraft design environment are used (e.g. wing aerodynamics). The fidelity levels of the analysis modules reach from simple scripts incorporating textbook formulas up to the integration of sophisticated numerical simulation tools (e.g. FEM).

The basic anatomy of analysis modules is always identical. They read all inputs from a CPACS dataset, prepare and execute an analysis, evaluate their results and finally write them back to the CPACS dataset and return the dataset. Most analysis modules integrate existing computational tools into the design environment. Figure 6 sketches the typical structure of such a wrapper module for the integration of the panel code VSAERO into the design environment. It uses utility libraries (TIXI, TIGL) described in the following section for all CPACS input and output operations.

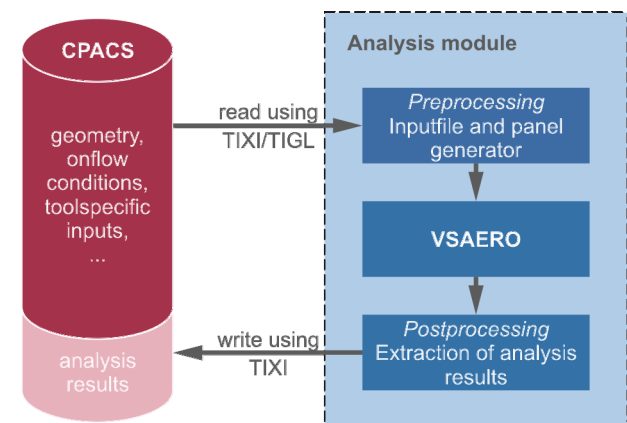


Figure 6: Typical structure of a CPACS wrapper module

The analysis modules are developed and maintained by disciplinary experts and run on analysis servers of the responsible institute. The tool developers also provide tech-

nical support for their analysis modules. The resulting clear responsibilities and small development teams are beneficial for the efficiency during tool development.

The RIDE analysis modules were developed with the overall rotorcraft design process (Figure 2) in mind. But the modular data-centric architecture of the design environment does also allow the rearrangement of modules and the quick creation of custom toolchains in order to address specific design tasks in a very flexible way.

2.4. Utility Libraries

As there are many recurring tasks in the development of analysis modules, the DLR developed a set of libraries, which provide standardized routines for these tasks. They simplify the analysis module development and reduce the risk of data misinterpretation. The following libraries are available:

- *TIXI* is a library for general CPACS Input/Output (IO) operations. It allows to easily access all elements in the XML data structure of CPACS datasets and to execute read and write operations.
- The graphics library *TIGL* builds an internal 3D CAD model of geometric components defined in CPACS datasets. Using that model it provides geometry query functions (e.g. point coordinates), surface/volume calculations and the export of watertight geometries to legacy CAD and mesh formats.

3. GEOMETRY GENERATION

The integrated approach of the RIDE toolchain combining conceptual and preliminary design revealed a gap in the geometry description when stepping from the conceptual design stage to the preliminary design stage:

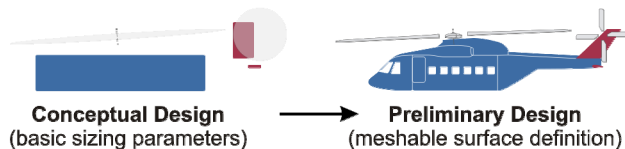


Figure 7: Gap in geometry description between conceptual and preliminary design

While the output of the conceptual design only includes basic geometric dimensions, some analysis modules (aerodynamics, structures) in preliminary design require detailed meshable surface definitions (see Figure 7). In order to close this gap, a new module for automatic geometry generation had to be developed.

3.1. Geometry Generation Concepts

Before starting with the implementation of the geometry tool, several available technologies and concepts found in literature about similar design frameworks and MDO toolchains have been assessed. Two basic approaches have been considered. The first one involved the integration of CPACS geometry output to the conceptual design module (similar to Böhnke et al.^[5]) and direct use of the parameters available in the CPACS dataset for geometry variation in preliminary design. The idea of the second approach was to incorporate alternative geometry parameterization and generation methods into the toolchain by automating and integrating an external tool and translating its geometry

output to CPACS. The following tools were considered:

- Easy-to-use conceptual geometry tools like NASA's Vehicle Sketch Pad (VSP)^[8] or ACBuilder^[20].
- Development of a new tool based on selected shape parameterization methods (e.g. CST^[9], B-Splines).
- Available commercial CAD systems with application programming interface (API) for automation. Examples for this approach can be found in papers by Ledermann et al.^[10], Rizzi et al.^[16] and Azamatov^[3].

The decision for one of these approaches has been made considering flexibility in geometry parameterization, user friendliness and implementation effort. Finally, a CAD-based approach using CATIA V5 has been selected, particularly because the available infrastructure, including geometry kernel and graphical user interface for creation of parametric component templates, minimizes the development effort for the core of a geometry generation tool. Further the straightforward creation and reparameterization of component templates using a well-known software environment provides a high level of flexibility and extensibility. Also, the generated geometry is available in a CAD format commonly used in subsequent steps of the design process. These advantages outweigh the drawbacks of the solution, including high license costs and lacking direct access to the geometry kernel.

3.2. Parametric Associative CAD Techniques

Modern CAD systems provide their users with a huge arsenal of features which can be used to build up "hierarchical parametric associative CAD models"^[10]. The construction of reusable, exchangeable components and assemblies demands a basic understanding of the available concepts, experience and careful design considerations in order to choose the right tool in the right order for each task. A brief overview of CAD techniques used in geometry templates for the newly developed module follows, including some CATIA-specific terminology.

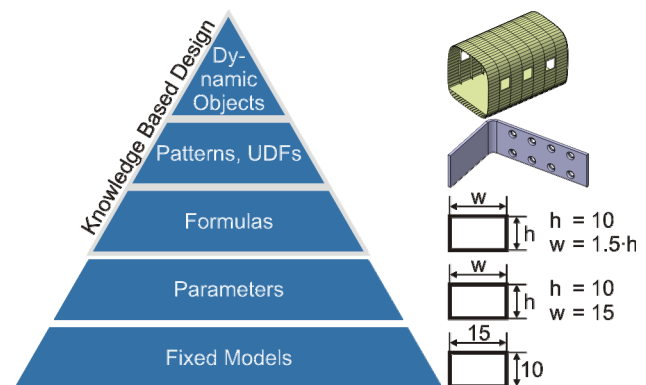


Figure 8: Parameterization levels used in modern CAD systems according to [10]

Figure 8 illustrates different levels of parameterization available in CATIA V5 according to Ledermann et al.^[10]. *Fixed models* don't use any parameters. All properties of geometric objects are stored in the respective geometric objects.

Parameters can be viewed as global variables in the namespace of CAD components or assemblies. They can

save information using various data types. The use of parameters is a way to store geometric properties outside of geometric objects. If a parameter is assigned to one or more geometric properties, it is possible to change all the geometric properties referring to the parameter at once by changing its value.

Going one step further, *formulas* can be used to define rules and to setup interrelationships between multiple geometric properties.

Patterns, User Defined Features (UDFs) and PowerCopies provide the user a possibility to define custom geometrical objects, which can later be instantiated multiple times. Patterns are used to create multiple instances of a previously defined object and arrange them in a pattern. They are instantiated dynamically, but all created objects will be exact clones of the referenced object and cannot be modified independently. Instances of UDFs and PowerCopies can be modified by changing their input parameters and geometries, but they are not instantiated dynamically.

Dynamic objects are a construct presented by Ledermann et al.^[10] which combines reactions and Visual Basic scripts to overcome the restrictions of patterns and UDFs. They can be used to dynamically create and adapt repetitive structures, like frames and stringers, to a given fuselage shape.

Formulas, patterns, UDFs and dynamic objects are ways to incorporate design knowledge and paradigms into CAD models.

Associativity is another common feature of modern CAD systems. It makes it possible to interrelate geometric objects by using references to existing geometric objects as inputs for the instantiation of new geometric objects. Using associativity the redundant definition of geometrical objects can be avoided. But attention must be paid to retain a hierarchy with a clearly defined data flow in order to avoid circular dependencies.

3.3. The RIDE Geometry Generation Module (GEOGEN)

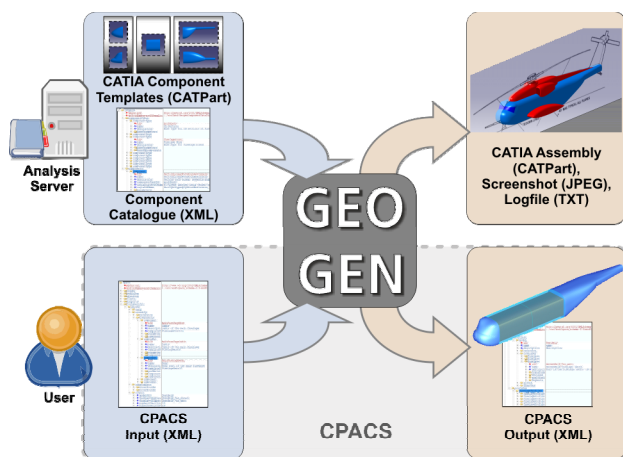


Figure 9: GEOGEN input and output files

The implemented geometry generation module (GEOGEN) is based on automation of the commercial CAD/CAE system CATIA V5 via its VBA (Visual Basic for Applications) programming interface. It basically instantiates components from a catalog of predefined templates, adapts their parameter values, assembles them and finally

exports the generated geometries to CPACS.

Figure 9 illustrates the files read and written by GEOGEN. Like all analysis modules of the RIDE design environment, it reads all input parameters from a CPACS dataset and writes the generated geometry back to this dataset. The component catalog is an XML database containing interface definitions and filenames of available geometry components. The generated assembly is also returned as a CATIA part document in native CATIA format (CATPart).

3.3.1. Component Templates and Script Components

Component templates and script components are the building blocks of all geometries generated using GEOGEN. They share the same general layout (Figure 10): A set of parameters and references to geometric components are the inputs used to define and return a new geometric component. The output geometry can be as simple as a point, but also a complex assembly of many geometric entities.

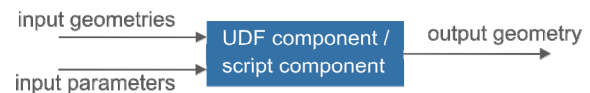


Figure 10: Inputs and outputs of component templates and script components

A component template is a UDF definition stored in a CATIA part file (CATPart). The geometry inside the UDF usually makes heavy use of parameters, formulas and other UDFs to integrate design knowledge into the geometry definition. Only basic CAD skills are required for the creation of component templates. CATIA's GUI is used for the construction of the geometry objects and for the definition of the UDFs including their interfaces (input parameters and geometries, outputs). Special attention must be paid to a robust geometry definition and a robust and efficient parameterization in order to avoid problems when instantiating the UDF with different input geometries and parameter values.

In some cases a desired functionality cannot be realized using UDFs, e.g. if a component should be used to join a variable number of input surfaces or in the use cases for dynamic objects described by Ledermann et al.^[10]. In such cases the functionality can be implemented using script components. They execute a Visual Basic script which generates the output geometry. The scripts can be arbitrarily complex and are only restricted by the functionality of CATIA's programming interface. I.e. script components can not only be used to generate predefined geometry objects, but can also incorporate operations and thus be used for the assembly of subcomponents. Programming skills are needed for the creation of script components, even though a great part of the code can be generated using CATIA's macro recorder.

3.3.2. Component Catalog

The component catalog is an XML database containing a list of all available components together with definitions of their interfaces. It is used by GEOGEN for the validation of the assembly definitions in the user inputs and to read the filenames and default parameter values of UDF and script components to be instantiated. Furthermore it is used for the automatic generation of a reference documentation of all available components, their interfaces and parameters.

This documentation serves the user as a starting point to create the assembly definition in the toolspecific inputs for GEOGEN.

The definition of component types and components in the component catalog resembles techniques known from object oriented programming languages:

- *Component type definitions.* Component types are used to define standardized interfaces for components and hence allow any component in an assembly to be exchanged by components of the same type. They correspond to abstract base classes in object oriented programming.
- *Parametric geometric components.* Components serve as a framework for the interface definition and the creation or instantiation of CAD geometries using the component templates or script components described in the previous section. Each component is assigned to a component type, thus inheriting all parameter and subcomponent definitions from this component type, corresponding to classes derived from abstract base classes in object oriented programming.

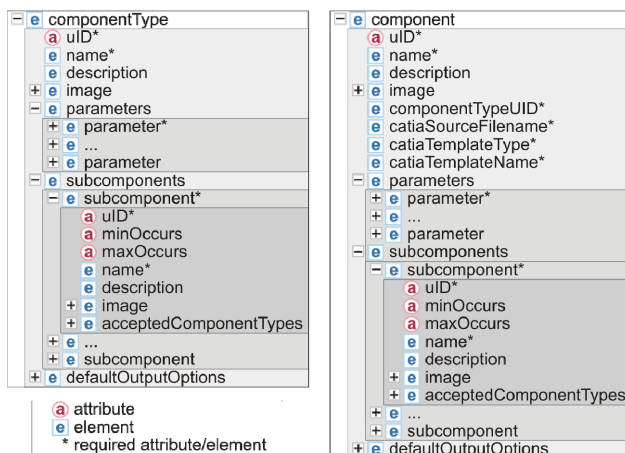


Figure 11: Structure of component types and components in the GEOGEN component catalog

Figure 11 depicts the structure of component type and component definitions. Only subcomponent and parameter elements are relevant for the interface definition. The other elements include metadata used to instantiate or execute the linked component templates and script components or to complement the information required for the generation of documentation for available component types, components, the associated subcomponents, parameters and their bounds.

A component catalog containing component type definitions and a set of basic components indented for the creation of rotorcraft geometries suited for preliminary design analyses has been created in the project RIDE. The generated components focus on the generation of conventional and tandem helicopter fuselage geometries, but also contain simple templates suited for the creation of stabilizer surface, rotor blade and wing geometries. Figure 12 shows example geometries generated using components from the RIDE component catalog.

Considerable work has been spent in the definition of component type interfaces, in order to realize a modular concept allowing to model a great spectrum of rotorcraft configurations by recombining and exchanging individual

components, and tweaking their parameters. Also the construction and parameterization of the generic geometric components has been a work-intensive task. The choice of a parameterization always implies a trade-off between flexibility and efficiency. A minimum number of parameters is desirable, but impossible to attain without restricting the design space. In addition, the parameterization should be intuitive and robust; the parameters should be meaningful and have clear bounds. Careful consideration during the construction process, ensuring that no errors occur when parameters are changed within their ranges or when using different input geometries, helps to increase the robustness of the templates.

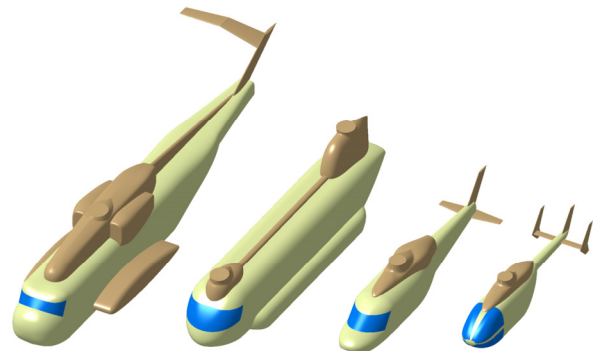


Figure 12: Example geometries generated using GEOGEN and the RIDE component catalog

The RIDE component catalog contains definitions for the following component types. The numbers indicate the number of currently available components:

- *Fuselage* (1)
 - *FuselageFront* (3)
 - *FuselageMid* (2)
 - *FuselageRear* (2)
 - *Tail* (2)
 - *RearCap* (3),
 - *FuselageAttachment* (e.g. engine cowlings, sponsons) (3)
- *RotorBlade* (1)
- *Wing* (1)
- *Profile2D* (5)
- *Point* (6)

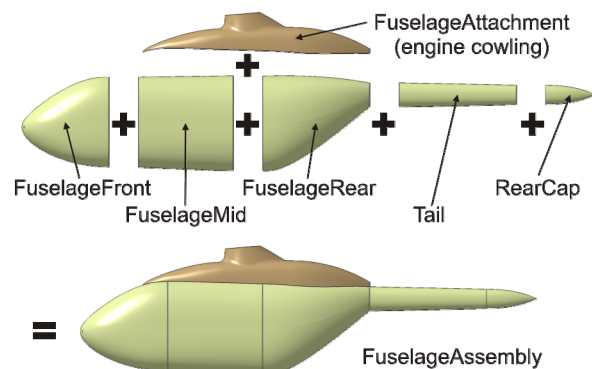


Figure 13: Example of a fuselage assembly using components from the RIDE component catalog

The fuselage component is the only one implemented as script component, because it can be instantiated with a variable number of subcomponents (e.g. multiple *Fuse-*

lageMid components). The associated script creates an assembly of the surfaces of all subcomponents (except those of type *FuselageAttachment*), converts the resulting watertight surface into a volume and merges this volume with the volumes of all subcomponents of type *FuselageAttachment*.

Figure 13 depicts an example fuselage assembly generated using components from the RIDE component catalog and its subcomponents.

3.3.3. Boundary Checks

GEOGEN includes an option to check whether the generated assembly satisfies geometric boundary conditions usually coming from the requirements definition. The boundaries are specified by CPACS fuselage definitions in the input dataset. Two boundaries can currently be specified (Figure 14):

- An *inner volume* along with a minimum offset, used for specification of the required cabin volume.
- An *outer volume* along with a maximum offset, used for specification of the maximal outer dimensions.

If a boundary check fails, GEOGEN returns a predefined exit code without exporting the geometry to the CPACS dataset.

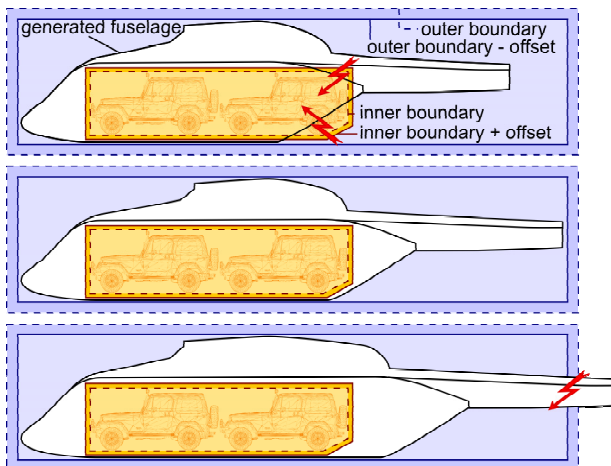


Figure 14: Boundary checks. Top: Inner boundary fails; Mid: OK; Bottom: Outer boundary check fails

3.3.4. Evaluation of Mass Positions

The evaluation of mass positions is a feature mainly intended for the use of GEOGEN within the RIDE toolchain. It allows the definition of rules for component mass positions, like engine, transmission and avionics positions, using point components from the component catalog. The RIDE component catalog contains point components for the definition of points relative to any component generated by GEOGEN, at area or volume centroids of components, at a ratio between two previously defined point components or using local component coordinate systems. The point components can also be combined, for example to define a point halfway between the volume centroids of two fuselage components. These point components are instantiated by GEOGEN and the absolute coordinates of the resulting points are written to the CPACS dataset. The outputs are intended for subsequent modules in the RIDE toolchain, e.g. for weight and balance calculations.

3.3.5. User Inputs

All user inputs for GEOGEN are located in the tool-specific node of the input CPACS dataset. Assembly definitions for fuselages, wings and rotor blades are the centerpiece of the GEOGEN user input. They are used to select the components from the component catalog to be instantiated and assembled, optionally set the values of their parameters and define their subcomponents. The user can define a global reference element and reference elements for every fuselage, wing and rotor blade assembly. If such reference elements are defined, all length/width/height type dimensions of all components and subcomponents will be multiplied with the respective reference length/width/height. This is used to streamline the workflow when coming from conceptual design: the conceptual design tool changes the reference values to the results of its sizing procedures and the geometry created by GEOGEN will be adapted to these dimensions. The user usually doesn't create a new assembly definition from scratch, but rather selects one of the available configuration templates and only replaces individual components and adapts some parameter values.

Besides the assembly definition, the tool-specific inputs also include definitions of boundaries (section 3.3.3), mass positions (section 3.3.4) and output options used for the export of the generated CAD geometries to CPACS.

3.3.6. Process Description

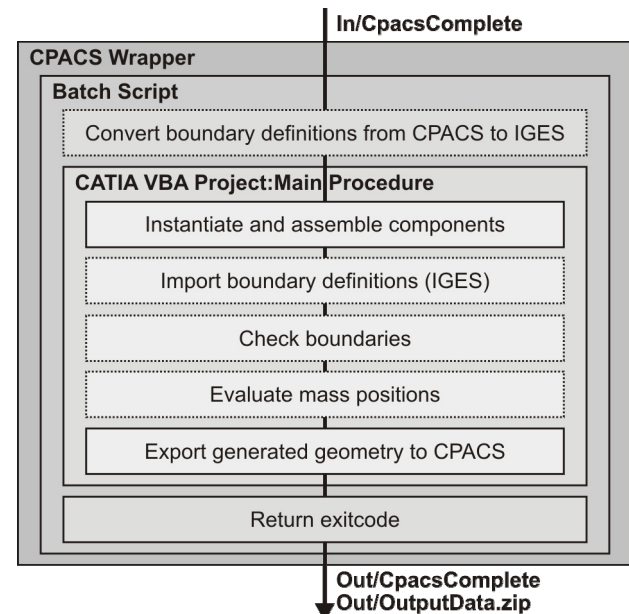


Figure 15: GEOGEN flow chart

Figure 15 visualizes the main steps of the GEOGEN process. GEOGEN is embedded in ModelCenter using a CPACSWrapper component. This wrapper calls a batch script, which first converts any geometric boundary definitions present in the CPACS dataset into surfaces and writes them to an IGES file using the TIGL library. Then the main procedure of the CATIA VBA project is executed. It starts with the automatic dependency aware instantiation and creation of all components and assemblies defined in the tool-specific inputs of the CPACS dataset. Next, it sets all parameter values of the created geometries to the values specified in the user inputs.

If boundary definitions are present in the CPACS dataset, it imports the geometries from the previously generated IGES file and checks if the boundary conditions are fulfilled (section 3.3.3). If the check fails, GEOGEN returns a predefined exit code and exits without exporting the geometry.

Next, all mass positions defined in the user input are evaluated (section 3.3.4) and the resulting absolute coordinates are written to the associated location elements of the CPACS mass breakdown.

Finally, the generated geometries are converted to CPACS and written into the CPACS dataset. Therefore a CATIA macro for the export of CATIA geometries to CPACS has been developed (CATIA2CPACS) and is invoked by GEOGEN.

All output files are then returned to the integration framework via the CPACSWrapper component.

4. FUSELAGE AERODYNAMICS

In the RIDE toolbox several modules of multiple fidelity levels have been developed for the evaluation of the aerodynamic performance of isolated fuselages. This paper focuses on the module AEROFUSE, as it is the only module, which considers the effect of local fuselage shape on the aerodynamic performance. AEROFUSE is based on VSAERO^[14], a commercial linearized 3D panel code with coupled viscous boundary layer calculations and a subsequent simple procedure for estimation of the pressure drag caused by separated flow.

Figure 16 outlines the complete AEROFUSE process. Its basic structure corresponds to the typical structure of a wrapper module, as shown in Figure 6 (section 2.3).

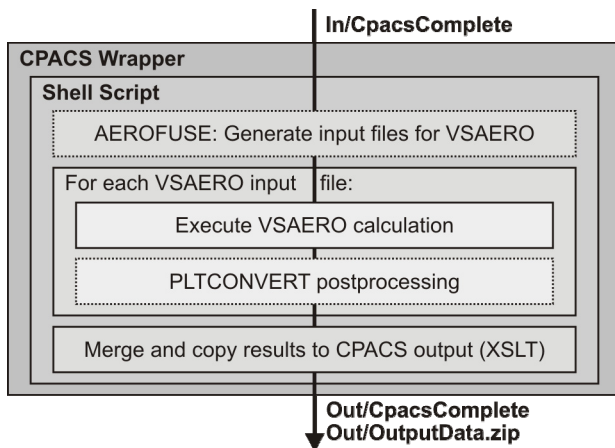


Figure 16: AEROFUSE flow chart

The CPACSWrapper calls a shell script which first executes the AEROFUSE main program. This is a C++ program, which first reads the input CPACS dataset using the TIXI library, then generates a fuselage surface grid using geometry query functions provided by the TIGL library and finally writes the fuselage panels to an input file for VSAERO along with additional data required for the aerodynamic analysis (e.g. onflow conditions, reference lengths). The density of the paneling is based on tool-specific user inputs. A maximum of 20 flow cases can be defined per VSAERO input file. If more onflow conditions are defined in the CPACS file, additional VSAERO input files are created. When the input files have been success-

fully generated, VSAERO is executed for each input file. Each VSAERO calculation is followed by a call of the newly implemented postprocessing tool PLTCONVERT. This tool reads total coefficients, surface data, streamlines and boundary layer data from the binary VSAERO output files, optionally starts a subroutine for the estimation of additional pressure drag due to separated flow (section 4.1), and writes the calculated total force and moment coefficients into an XML file. When all calculations are completed, an XSLT transformation merges the results of the individual XML output files into an *aeroPerformanceMap* element in the analysis branch of the CPACS file. Finally, the CPACS file and a ZIP archive containing logfiles and optional VSAERO and PLTCONVERT output files are returned to the integration framework via the CPACSWrapper.

4.1. Pressure Drag Estimation

Helicopter fuselage design is often driven by operational and functional considerations, while the aerodynamic efficiency only plays a secondary role. This leads to bluff fuselage shapes, whose flowfield is typically dominated by a region with separated flow at the aft of the fuselage, even in the cruise design point. Hence the drag component caused by separated flow cannot be neglected when considering helicopter fuselage drag.

Potential codes do not account for the viscous pressure drag caused by separation when only using body surface panels. Using VSAERO, the effects of flow separation are usually modeled by manual definition of wake panels shedding at the separation line. But tests showed that this procedure cannot be easily automated and lacks robustness. Thus a simpler, more robust method for the estimation of viscous drag has been implemented in the post-processing tool PLTCONVERT using the surface pressure and boundary layer data output by VSAERO. This method is based on the assumption of constant pressure in areas of separated flow.

The experimental averaged surface pressure distribution on a sphere in Figure 17 substantiates this assumption for bluff bodies, as the surface pressure remains almost constant starting from the flow separation point. However, the implemented pressure drag correction neglects the effect of the flow separation on the pressure distribution upstream of the separation point.

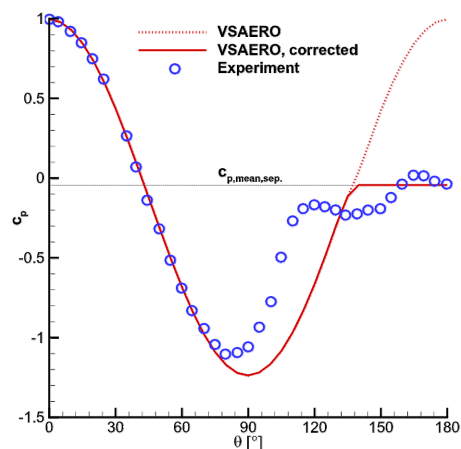


Figure 17: Pressure distribution on a sphere at $Re=1.1 \times 10^6$: averaged experimental^[11] and VSAERO results

The assumed value for the pressure coefficient in these areas can either be set by the user or determined automatically (default). In the latter case it is derived by calculating the mean pressure coefficient on the separation line predicted by the boundary layer procedures in VSAERO. The effect of the viscous pressure drag on the global force and moment 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 ($c_f \leq 0$).

5. APPLICATION EXAMPLES

In this section the use and the functionality of the presented analysis modules are demonstrated. First, the aerodynamics module AEROFUSE is applied to testcases with generic fuselage geometries and its results are compared to available wind tunnel test data and CFD results. Then a parametric study is performed showing the influence of the fuselage shape on its aerodynamic performance by coupling the geometry module GEOGEN with the fuselage aerodynamics module AEROFUSE in the integration framework ModelCenter.

5.1. ROBIN and ROBIN-mod7 Testcases

The well-known ROBIN fuselage^[13] has been selected as a first testcase for the AEROFUSE module, because an exact geometry description of the generic fuselage shape is available along with experimental pressure data.

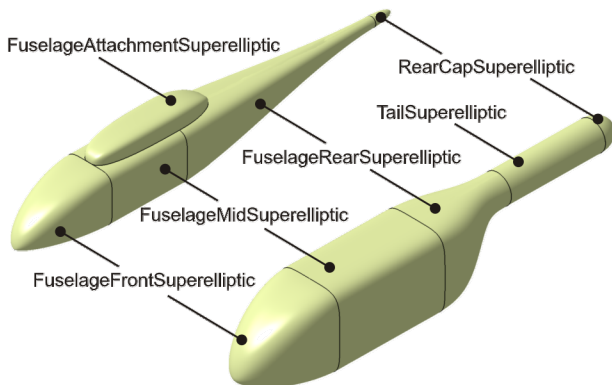


Figure 18: New components added to the GEOGEN component catalog to build ROBIN-like fuselages: original ROBIN and ROBIN-mod7

To obtain comparable results in aerodynamic analyses, it is crucial to use accurate surface representations for the body surfaces. Therefore in a first step the analytical description of the ROBIN fuselage and pylon has been incorporated into component templates and added to the GEOGEN component catalog, thus making the parameterization of the fuselage available in the RIDE design environment. Figure 18 shows assemblies of the ROBIN^[13] fuselage and the derived ROBIN-mod7^[19] body in CATIA, generated using the newly implemented component templates based on analytical super-elliptical cross-section definitions.

Then the ROBIN geometry has been exported to a CPACS dataset using GEOGEN. This dataset was used as input for the aerodynamic evaluation with the AEROFUSE module. The onflow conditions were set according to the wind tunnel testcase^[13].

α : 0°
 Ma : 0.062
 Re : 4.46×10^6

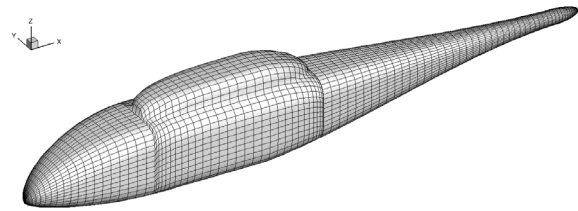


Figure 19: Paneling used for the ROBIN testcase

The paneling was set to 100 panels in streamwise direction with points distributed in three zones: using a half cosine distribution near the nose until the begin of the pylon, a cosine distribution in the pylon region and a half cosine distribution for the fuselage aft behind the pylon with higher density at the end of the fuselage (see Figure 19). The paneling in circumferential direction was set to use 60 equidistantly distributed points. 250 streamlines were distributed on the fuselage for use of the boundary-layer analysis in VSAERO. The number of viscous iterations was set to 2, default values were used for the boundary layer analysis options.

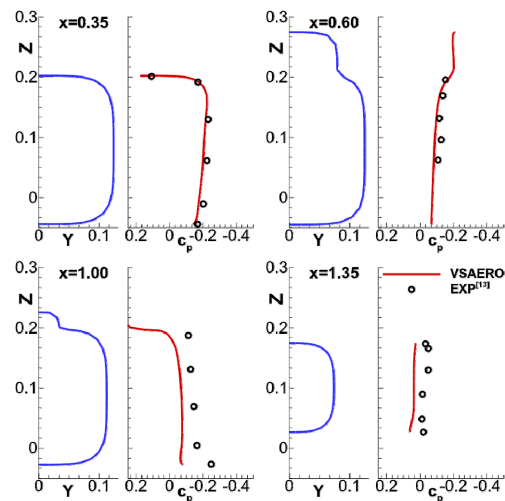


Figure 20: Comparison of VSAERO surface pressure results for the ROBIN testcase with experimental results from^[13]

Figure 20 shows the results of the VSAERO analysis using AEROFUSE. The surface pressure predicted agrees very well with the experimental data, particularly in the front region of the fuselage. This shows that panel codes are well suited for the application to flow problems around streamlined shapes, like the ROBIN fuselage, where almost no separation occurs in the flowfield.

Unfortunately there was no experimental total drag coefficient data available for the ROBIN testcase and the minimal area of separated flow did not provide a good benchmark for the separation drag estimation routine used in AEROFUSE. Therefore the baseline testcase presented in^[19] was selected for a further application of AEROFUSE. The generic ROBIN-mod7 fuselage shape is based on the ROBIN parameterization. The bluff aft-body shape of the fuselage is a good test for the separation drag estimation routine used in AEROFUSE. The geometry parameters are available along with experimental data and CFD re-

sults in ^[19].

The ROBIN-mod7 geometry was made available to AEROFUSE using the procedure described for the ROBIN fuselage. The onflow conditions were set according to the baseline case in ^[19]:

α : 0°
 Ma: 0.1
 Re: 1.6×10^6

The paneling was set to 100 panels in streamwise direction with points distributed using a cosine distribution and 64 panels in circumferential direction using equidistant points. The streamlines and the number of viscous iterations were setup similar to the ROBIN testcase. Furthermore, transition was tripped right at the fuselage nose (Figure 22) for the boundary layer analysis in order to achieve results comparable to the tripped wind tunnel flow and the fully turbulent RANS calculations in ^[19]. Figure 21 shows the resulting centerline surface pressure distribution as calculated by VSAERO and with applied correction using the implemented pressure drag estimation routine in AEROFUSE.

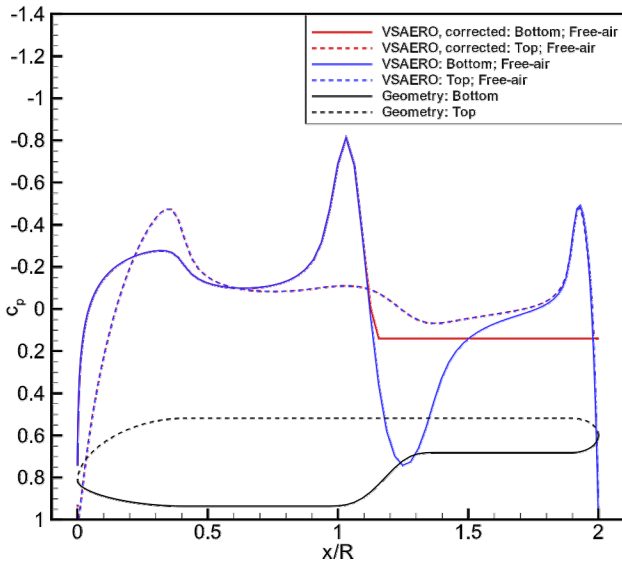


Figure 21: Center line surface pressure distributions for the ROBIN-mod7 testcase using VSAERO with and without the implemented pressure drag estimation method ($\alpha=0^\circ$, $Ma=0.1$)

A similar figure in ^[19] shows that the viscous flow separation considerably affects the pressure distribution along the fuselage bottom line, even upwind of the separation line. This effect cannot be captured by VSAERO without wake modeling. While the CFD results suggest, that the flow reattaches at the tail at $x/R \approx 1.5$, the VSAERO boundary layer analysis assumes that the flow stays separated along streamlines downwind of the predicted turbulent separation points (Figure 22). Nevertheless the separation line predicted by VSAERO matches the experimental results in ^[19] well.

As expected, the pressure distribution calculated by VSAERO for the streamlined top of the fuselage agrees very well with the CFD results until separation occurs at the very aft of the fuselage.

The total drag coefficients in Table 1 calculated using the RANS code OVERFLOW in ^[19] show the huge influence of the wind tunnel walls and model support on the total drag.

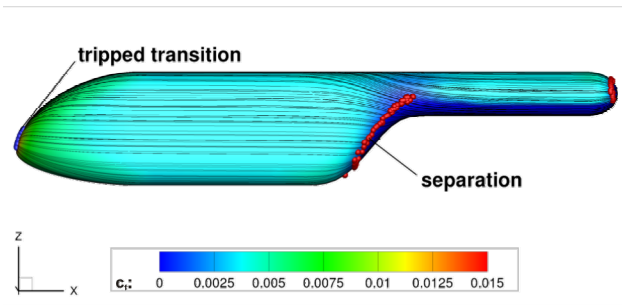


Figure 22: Friction coefficient and separation points calculated by VSAERO

$\alpha = 0^\circ$		$C_{D,visc}$	$C_{D,press}$	$C_{D,tot}$
OVERFLOW, SA ^[19]	tunnel	0.055	0.090	0.145
OVERFLOW, SA ^[19]	free-air	0.056	0.058	0.114
AEROFUSE	free-air	0.050	0.101	0.151
EXPERIMENT ^[19]				0.145

Table 1: Comparison of calculated drag coefficients and their components with CFD results and experimental data from ^[19] for $\alpha=0^\circ$

$\alpha = -5^\circ$		$C_{D,visc}$	$C_{D,press}$	$C_{D,tot}$
OVERFLOW, SA ^[19]	tunnel	0.057	0.091	0.148
OVERFLOW, SA ^[19]	free-air	0.056	0.066	0.122
AEROFUSE	free-air	0.043	0.107	0.150
EXPERIMENT ^[19]				0.153

Table 2: Comparison of calculated drag coefficients and their components with CFD results and experimental data from ^[19] for $\alpha=-5^\circ$

The OVERFLOW simulations including the wind tunnel agree very well with the experimental results. Therefore the free-air OVERFLOW simulation results are used as a reference for the drag estimated using AEROFUSE. Compared to these, AEROFUSE overpredicts the total drag by 32%. The difference is mainly caused by the estimated separation drag. The steep c_p gradient near the separation point makes the assumed c_p in the separated flow region extremely sensitive to the predicted separation position, thus affecting the estimated drag due to separation. Moreover the velocities induced by the vorticity of the separation wake aren't considered by the simple separation drag estimation routine and the influence of the separation on the pressure distribution upwind of the separation line is also neglected. Finally, the reattachment of the flow at the tail is also not considered. This is probably the reason for the lower viscous drag predicted by VSAERO, as it assumes zero friction downstream of the separation line.

The complex flowfield around bluff helicopter bodies is very difficult to predict using simplified models. Considering that the total drag results using different high-fidelity RANS codes and turbulence models presented in ^[19] vary by up to 25%, the accuracy of the presented drag estimation method is satisfactory. While it should not be used for a reliable prediction of exact total coefficients, the method can be useful for preliminary studies for the minimization of pressure drag, as its accuracy grows when the area of separated flow decreases. The method can also be used to quickly obtain qualitative results when comparing different fuselage configurations only requiring computation time in the order of seconds on modern desktop PCs, while high fidelity RANS simulations require computation times in the order of hours, days or even weeks.

5.2. Parametric Study

A simple toolchain combining multiple analysis modules in ModelCenter has been setup in order to show the typical workflow using the RIDE design environment. The aim of the presented toolchain is to perform a parametric study showing the effect of fuselage geometry variations on its predicted aerodynamic performance. The fuselage shape is slowly morphed starting from the ROBIN body-only geometry^[13] into the ROBIN-mod7 geometry by linearly interpolating between the coordinates of the two fuselages. Therefore the component templates shown in Figure 18 have been extended to return geometries generated by linearly interpolating between the coordinates of two fuselage sections defined by a ROBIN-like parameterization. An additional parameter a is used as the coefficient for the linear interpolation. $a=0$ returns a shape generated using the parameters of the first fuselage, $a=1$ returns a shape generated using the parameters of the second fuselage and $a=0.5$ returns the surface defined by all midpoints between the first and the second fuselage.



Figure 23: Toolchain for the parametric study in ModelCenter

Figure 23 outlines the toolchain created for the parametric study in ModelCenter. It uses a CPACSSource utility module to load an initial CPACS dataset into the toolchain. This dataset contains only a minimal set of required CPACS elements along with toolspecific inputs for GEOGEN and AEROFUSE. The GEOGEN inputs include an assembly definition using the newly created components.

A RealMerger component is used to write the actual value of the interpolation coefficient a from ModelCenter to the CPACS dataset. Then the incorporated analysis modules GEOGEN and AEROFUSE are executed. Finally the drag components calculated by AEROFUSE are read from the CPACS dataset into ModelCenter variables using a MultipleRealSplitter component.

Figure 24 summarizes the results of the analyses using ModelCenter's parametric study tool to vary the parameter a from 0 to 1 in steps of 0.1. All settings for the AEROFUSE module, including onflow conditions and paneling density, have been taken over from the ROBIN-mod7 testcase described in the previous section.

As expected, the predicted separation drag for the streamlined ROBIN body is negligible, but growing fast as the fuselage shape gets less streamlined. The jumps in the pressure drag curve are assumed to be caused by the relative large panel sizes in the region of the separation line and the fact that panels are treated in a binary way (either assigned to the area of separated flow or not) when summing up the force deltas.

The trend of the predicted viscous drag component also seems plausible. Its constant decrease corresponds to the decrease of the wetted-surface to frontal-area ratio. Furthermore the growing area of predicted separated flow doesn't contribute to the calculated viscous drag because VSAERO assumes zero friction in this area.

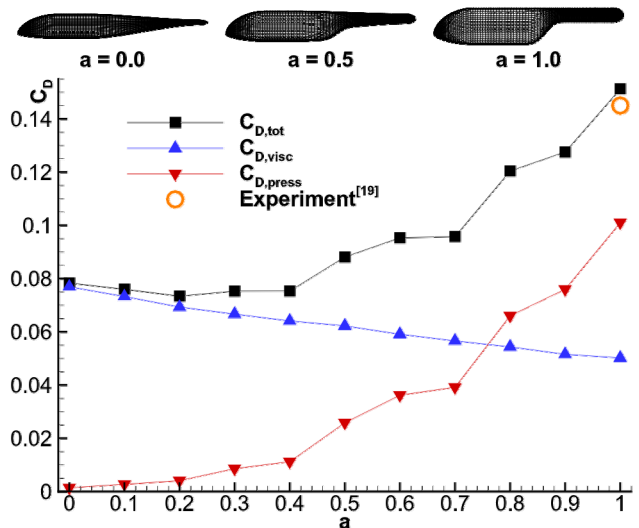


Figure 24: Result of the parametric study: estimated viscous, pressure and total drag coefficients as a function of the linear interpolation coefficient a

6. CONCLUSION AND OUTLOOK

After shortly reviewing the general architecture of the RIDE rotorcraft design environment, this paper focused on the implementation and functionality of two analysis modules for preliminary design developed at the Institute of Aerodynamics and Flow Technology: the geometry module GEOGEN and the fuselage aerodynamics module AEROFUSE.

GEOGEN provides a framework for automatic instantiation and assembly of predefined CAD templates with integration into the CPACS workflow. It can be used to automate the process of geometry generation to a great extent and thus close the gap between geometry representations in conceptual and preliminary design. A basic set of geometry component templates for GEOGEN has been generated within the RIDE project. They allow the user to easily assemble parametric geometries for a broad range of common rotorcraft configurations. Due to its modular design, geometry generation is not restricted to the available component templates, but can easily be extended and adapted to new tasks by incorporating new component type definitions and new component templates into the component catalog. Plans for the further development of GEOGEN include an assessment of possibilities to speedup its runtime, refining the geometry models generated in CATIA and exported to CPACS (e.g. adding windows and doors) and the adaption of the export process to upcoming CPACS features for the definition of smooth fuselage surfaces.

AEROFUSE is based on the commercial panel code VSAERO combined with a simple method for the prediction of viscous pressure drag. Good results were obtained using the method for drag prediction of streamlined fuselages dominated by viscous drag. The application example comparing AEROFUSE with experimental and RANS results for the ROBIN-mod7 testcase shows that an exact prediction of absolute aerodynamic forces using simple models is difficult if the flowfield contains large areas of separated flow, as the pressure drag estimation method

does not rely on physical models. However, considering the scattering of higher-fidelity RANS solutions for such testcases and the considerable savings in computation time, the deviation of the results obtained using the presented method from the experiment seems acceptable. The implemented module can be a useful tool for qualitative analyses or quick parametric studies with the aim to minimize or to eliminate the area with separated flow.

Finally, such a parametric study, coupling the modules GEOGEN and AEROFUSE in a simple toolchain in order to show the effect of fuselage shape on the predicted fuselage drag, has been conducted. The predicted trends seem plausible and an advance compared to common conceptual design methods, which are not able to model the effect of local shape on fuselage aerodynamics.

Next, the presented modules will be applied to preliminary design studies involving the complete RIDE toolchain. At the same time, the development of the analysis modules will be continued in an ongoing process in order to improve their functionality and robustness and to adapt them to new tasks.

REFERENCES

- [1] Achenbach, E.: *Experiments on the Flow Past Spheres at Very High Reynolds Numbers*. J. Fluid Mech., 54(3), pp. 565-575.
- [2] Anderson, J. D.: *Aircraft Performance and Design*. McGraw-Hill Education, 1999.
- [3] Azamatov, A., Lee, J.-W., Byun, Y.-H.: *Comprehensive aircraft configuration design tool for Integrated Product and Process Development*. Advances in Engineering Software, Vol. 42, Issues 1–2, January–February 2011, pp. 35-49.
- [4] Bachmann, A., Kunde, M., Litz, M., Schreiber, A., Bertsch, L.: *Automation of Aircraft Pre-design Using a Versatile Data Transfer and Storage Format in a Distributed Computing Environment*. Third International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2009), 2009.
- [5] Böhnke, D., Nagel, B., Gollnick, V.: *An Approach to Multi-Fidelity in Conceptual Aircraft Design in Distributed Design Environments*. IEEE Aerospace Conference, 2011.
- [6] Böhnke, D.: *Common Parametric Aircraft Configuration Schema (CPACS)*. v2.1, <http://software.dlr.de/p/cpacs>
- [7] Dassault Systèmes: *CATIA V5 software*. <http://www.3ds.com/products/catia>
- [8] Hahn, A.: *Vehicle Sketch Pad: Parametric Geometry for Conceptual Aircraft Design*. 48th AIAA Aerospace Sciences Meeting, Orlando, FL, Jan 4 - 7 2010, AIAA-2010-657.
- [9] Kulfan, B.M., Bussoletti J.E.: *"Fundamental" Parametric Geometry Representations for Aircraft Component Shapes*. AIAA-2006-6948, Sept. 2006
- [10] Ledermann, C., Hanske, C., Wenzel, J., Ermanni, P., and Kelm, R.: *Associative parametric CAE methods in the aircraft pre-design*. Aerospace science and technology, Vol. 9, No. 7, 2005, pp. 641-651.
- [11] Lier, M., Harbig, K., Kohlgrüber, D., Krenik, A., Kunze, P., Lützenburger, M.: *Studies on Rotorcraft Integrated Design and Evaluation at DLR - First Results*. 38th European Rotorcraft Forum, Amsterdam, Netherlands, 04.-07.09.2012.
- [12] Liersch, C. M., Hepperle, M.: *A Distributed Toolbox for Multidisciplinary Preliminary Aircraft Design*. CEAS Aeronautical Journal, Vol. 2 (Number 1-4), pp. 57-68.
- [13] Freeman, C., Mineck, R. E.: *Fuselage Surface Pressure Measurements of a Helicopter Wind-Tunnel Model with a 3.15-meter Diameter Single Rotor*. NASA TM 80051, Langley Research Center, March 1979.
- [14] Nathman, J. K.: *VSAERO – A Computer Program for Calculating the Nonlinear Aerodynamic Characteristics of Arbitrary Configurations*. User's Manual, Version 7.2, Analytical Methods, Inc., September 2007.
- [15] Phoenix Integration: *ModelCenter Software*. www.phoenix-int.com/software/phx-modelcenter.php
- [16] Rizzi, A., Berard, A., Isikveren, A.T.: *CADac: a new geometry construction tool for aerospace vehicle pre-design and conceptual design*. 26th AIAA Applied Aerodynamics Conference, Honolulu, HI, USA, August 18–21, 2008, AIAA-2008-6219.
- [17] Rizzi, A., Zhang, M., Nagel, B., Böhnke, D., Saquet, P.: *Towards a Unified Framework using CPACS for Geometry Management in Aircraft Design*. 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, AIAA 2012-0549, 2012.
- [18] Roskam, J.: *Airplane Design*. DARCorporation I-VII, 1989.
- [19] Schaeffler, N.W., Allan, B. G., Lienard, C., and Le Pape, A.: *Progress Towards Fuselage Drag Reduction via Active Flow Control: A Combined CFD and Experimental Effort*. 36th European Rotorcraft Forum Paper 064, Paris, France, September 2010.
- [20] Tomac, M., Eller, D.: *From geometry to CFD grids – An automated approach for conceptual design*. Progress in Aerospace Sciences, Vol. 47, Issue 8, November 2011, pp. 589-596.
- [21] Torenbeek, E.: *Synthesis of Subsonic Airplane Design*. Delft University Press, 1982.