AE4-233 Advanced Design Methods 2014/2015

Part 2. KBE Homework exercise

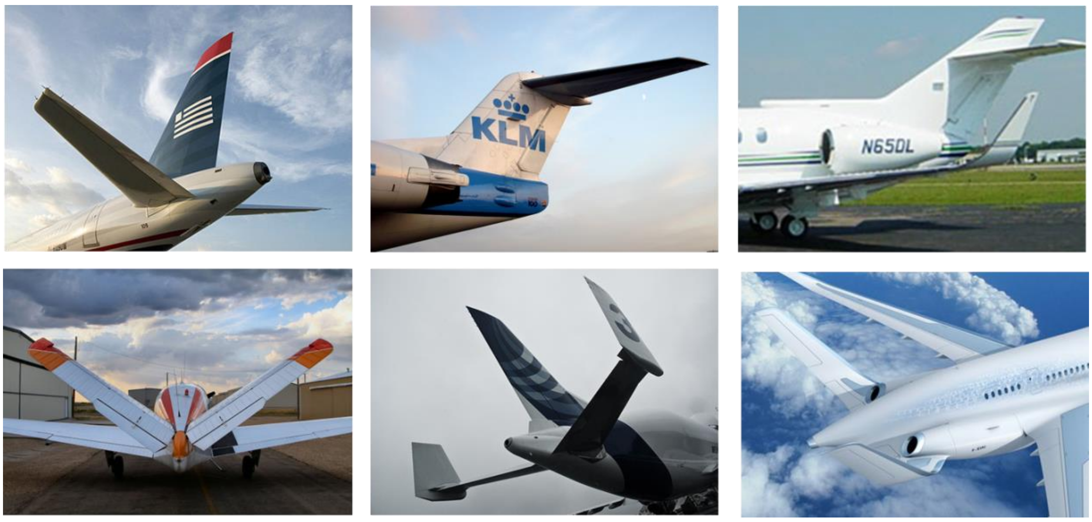
Generation of a KBE application to support jet aircraft tail design

The scope of this assignment is the development of a KBE application to support the geometry modeling and analysis of various aircraft configurations, with particular focus on the design of the tail. Although simple, this application will feature some of the characteristics of a typical KBE tool for industrial use, i.e. it will:

- Provide a parametric rule based model of the product under consideration,
- Read input data from external files,
- Make use of rules to generate a design that complies to best practice and regulations,
- Extract information from the product model to feed internal and/or external calculation modules and simulation tools,
- Generate various output files (e.g. PDF, IGES, STEP) containing calculation results and geometry data to be exported to other CAD/CAE systems.

The design case

SmartWorks B.V., the engineering company where you and your team mate are working as knowledge engineers, has a number of active collaborations with different aircraft manufacturers, such as Airbus, Boeing, Dassault, Embraer, etc., concerning with the development of software tools to support the conceptual and preliminary design of transport aircraft. Large part of the conceptual design carried out by the abovementioned manufacturers is still performed by means of “pen and paper”, spreadsheets and a lot of manual work behind a CAD system. This design approach is very flexible and gives designers a lot of freedom. However, it is far from efficient and has several limitations. For example, the lack of automation prevents using optimization techniques early in the design process. Furthermore it limits the extent of the design space that can be explored in the limited available time and creates a lot of frustration for CAD and CAE specialists, who need accurate data and consistent



models to proceed with the design process.

SmartWorks B.V. is considering the **development of a KBE application to support the conceptual design phase of jet aircraft, with particular focus on the sizing and analysis of the tail**. The main objective of the tool is to deliver, in a very short time, models of passenger jet aircraft with automatically sized tailplanes, supplemented by basic performance calculations and dedicated geometry representations to perform further simulations and configuration optimization.

Before committing to the full development of this KBE application, SmartWorks wants to develop first a **demonstrator tool** and shows its potential benefits to their customers. Being a demonstrator, limited resources are available for its development; hence, the tool capabilities in terms of modeling fidelity, flexibility, user interface, etc., must be kept to a "low level". To this purpose the lead engineering team of SmartWorks B.V. has specified the *use case* for the demonstrator (i.e. what the demonstrator is supposed to do), which you can read in the next section.

Now it is up to you and your team mate to develop, in the little time available, the best possible KBE application to convince your management to commit to the full development of this aircraft modeling and tail sizing tool!

Requirements for the KBE demonstrator

Starting from a basic set of input data (e.g., fuselage length and diameter, wing planform, number and position of engines, etc.) and making use of the design rules and methods provided as annex to this assignment[1], a KBE demonstrator tool will be developed, with the following capabilities and functionalities:

Rule-based parametric modeling functionalities

1. Modeling of complete transport aircraft configurations.

- The tool will have sufficient flexibility to allow a quick generation of various aircraft models, ranging from small regional jets to very large airliners. Both high and low wing configurations must be generated. Both wing- and fuselage-podded engine configurations - with 2, 3 or 4 engines - must be generated.
- The tool modeling capability will be adequate to address conventional aircraft configurations with (quasi-)cylindrical fuselage, cantilever wing and after tail.
- The use of simple cylinders and/or cones is acceptable for modeling parts of the fuselage and the engine nacelles. However, the fuselage nose and tail cone sections will have to be modeled using realistic curved surfaces, parameterized in such a way to allow the implementation of various slenderness ratios, divergence and rotation angles.
- The wing will be modeled using realistic airfoils. The user must be allowed to define the wing planform shape by means of top level parameters such as, span, aspect ratio, taper, sweep, kink position, chord lengths, etc.
- Concerning the modeling of the tail, which is the core functionality of the tool, the generation of the following tail configurations must be guaranteed:
    - **Conventional tail**: both horizontal and vertical tail planes attached to the fuselage
    - **T-tail**: horizontal tailplane on top of the vertical tailplane
    - **Cruciform tail**: horizontal tail plane located somewhere between the root and the tip of the vertical tailplane
    - **H-tail/C-tail**: two vertical tailplanes located at the tip of the horizontal tailplanes
    - **V-tail**: two tailplanes set at adequate dihedral angle, integrating the functions of both horizontal and tailplanes.
- The tailplane will be modeled using realistic airfoils
- Modeling of the fuselage interiors, aircraft structure and landing gear are outside the scope of the demonstrator. Details such as doors, cockpit and cabin windows, cutouts, fairings, pylons, control surfaces, high lift devices, etc., can also be neglected.

Input data handling capabilities

1. The tool shall read all the required input data from **external input files, defined in neutral format** (i.e. not written according to the syntax of the used KBE language, but as plain text files without parenthesis, columns, etc.)

[1] Some selected reference material (design rules, methods and data) is provided on BB. More detail in the section *Provided Reference Material* later in this document.

2. The tool shall allow editing the main input design parameters also **interactively**, by means of **settable slots** in the KBE development interface (Tasty)
3. The airfoils used to model wing and tailplanes will be generated on the basis of predefined airfoils data, stored in dedicated .dat files. These airfoils will be expressed as sets of 2D Cartesian coordinates with chord length equal to 1, or as sets of CST coefficients. You are free to choose one of these two representations, although the CST option will facilitate the generation of one of the output described below, in the *Capabilities to support external analysis tools* section)
4. When possible, the tools shall **provide warnings** if wrong or inconsistent input values are provided (e.g., outside range values, or values leading to unfeasible or inconsistent geometrical design) **and** overwrite wrong input values with correct ones (still informing the user in case of some value overwriting)

Preliminary sizing and analysis capabilities

The tool will feature the following automatic sizing capabilities (consult the reference material packages for the necessary sizing methods)

1. A suitable longitudinal **wing position** will be automatically determined by means of simple rules that account for the aircraft engine configuration (i.e., wing or fuselage podded)
2. The **sizing of the tail** (planform area, shape, thickness, etc.) will be performed by means of the **tail-volume coefficient approach** and according to the design rules and data provided in the reference material package
3. The length and spanwise position of the **Mean Aerodynamic Chord** (**MAC**) of the wing and all the tailplanes will be computed. The MAC will be graphically displayed in the aircraft geometry model.
4. The relative positioning of horizontal and vertical tailplanes will guarantee that a sufficient portion of the **rudder is not blanketed by the horizontal tailplane** wake, at high angle of attack. The tool will provide **visual evidence** of this constraint and a numerical value of the percentage area of the un-blanketed rudder
5. The position of the horizontal tail with respect to the wing will be checked against the **risk of deep stall** (see graphical method in the provided reference material). The tool will provide **visual evidence** of this constraint

The tool will compute the following values (consult the reference material packages for the necessary methods):

1. the **weight of the tailplanes** using a Class II weight estimation method
2. the **reference area**, the **exposed area** and the **wetted area** of wing **and** tailplanes, using the generated 3D geometry models
3. The **lift gradients** (computed at cruise conditions) of:
    - wing ($\mathbf{CL_{alpha\_w}}$)
    - fuselage-wing combination ($\mathbf{CL_{alpha\_wf}}$)
    - horizontal tailplane ($\mathbf{CL_{alpha\_h}}$)
4. The **wing downwash** gradient on the tail (computed at cruise conditions)
5. The **tailless aircraft**[2] **aerodynamic center** Xac (at cruise condition), including the contribution of wing, fuselage, nacelles and jet stream. The position of the aerodynamic center of the wing can be assumed at 0.25MAC for simplicity. The KBE tool will show the position of the aerodynamic center of the tailless aircraft as a point in the aircraft geometry model
6. The allowed **most aft position of the c.g.** which still complies with the longitudinal static stability requirement (use a stability margin value equal to 0.05MAC). The KBE tool will show the position of the most aft allowed c.g. position as a point in the aircraft geometry model

[2] This is the aerodynamic center of the whole aircraft without the tail contribution

Capabilities to support external analysis tools

The KBE application tool shall feature the following capabilities to support external analysis and optimization:

1. The tool will be able to interact with the open source 2D flow solver XFOIL (http://web.mit.edu/drela/Public/web/xfoil/)
   - It will send XFOIL the data of a lifting surface section to analyse. The user must be able to specify any 2D section, either from the wing or any tailplane (make sure to define the wing section to be analysed perpendicularly to the given lifting surface leading edge)
   - It will trigger the XFOIL solver
   - It will retrieve the analysis result and **plot the Cl-alpha** curve of the given 2D section in the KBE development environment (Tasty). It will also extract from the analysis result the value of maximum lift coefficient **Clmax** and relative critical angle of attack **Alpha_stall**
2. The tool will be able to generate a functioning **input file for the Q3D Aero Solver** (the code used in the MDO module of the course). Find an example of Q3D input file in the MDO Tutorial 3 folder. Note that this functionality implies the definition of the **2D sections by means of the CST method**.

Reporting capabilities

For any generated aircraft model, the tool will be able to output the following reports:

1. one PDF file including the top, side, front and trimetric **views** of the complete aircraft
2. one PDF file including:
   - the plot of the selected wing (or tail) 2D section for XFOIL analysis,
   - the relative Cl-alpha plot generated within the KBE application
   - the value of Clmax and Alpha_stall
3. One PDF file with a table containing all the relevant **input** used to generated the given aircraft instance and all the **computed** data listed in the *Preliminary sizing and analysis capabilities* section, including the values of the tailplanes geometrical parameters (do not forget to provide units!)
4. An **IGES** or **STEP file** of the generated aircraft, in order to allow importing the geometry model generated by the KBE application into a CAD tool of choice (e.g., CATIA)

NOTE that the tool should allow the user to trigger the generation of the PDF and IGES/STEP output files, **directly from Tasty** (e.g., after the interactive modification of some settable slot)

DELIVERABLES

DELIVERABLE 1 - to be uploaded on BB in digital version:

The working GDL application plus a *readme* file containing the very basic instructions to operate it (e.g. what to compile, name of the root object to instantiate, etc.).

Notes:

1. Use **plenty of comments and annotations** both inside your code and input files.
2. Organize your code in several .lisp files (e.g. one file per main class definition).

DELIVERABLE 2 - to be uploaded on BB in digital version <u>and</u> brought in 2 printout copies at the live demo session:

Collection of the following items:

1. One or more (if required for clarity) **UML class diagrams** representing the inheritance and part-whole relationships of the KBE application. Include the main classes and for each class report the main attributes. Make sure to use the GDL-oriented UML style presented during lecture (i.e. the UML diagrams with the *<<child-object>>* blocks and "*of type*" connectors)

2. The various automatically generated **PDF reports** (see *Reporting capabilities* section) for **one** design instance

3. A copy of the automatically generated **Q3D aero solver input file with relative output** file for **one** design instance

4. **Screenshots** from CATIA (or some other CAD system of your choice), showing the geometry models of **one** design instance, imported from the automatically generated IGES/STEP files (see *Reporting capabilities* section).

ASSESSMENT

The grade for your KBE module will depend on the quality of the two abovementioned deliverables, which will be assessed by a team of KBE experts[3], during a live demonstration**.** Each team will sign up to one of the scheduled **1 hour live demonstration slots** (refer to BB for the schedule of the live demonstration sessions).

The live demo session is organized in two main parts:

1. In the **first part** (about 20 minutes), you will **present, discuss and demonstrate** all the functionalities (and limitations) of your KBE tool. Within this time, you can make use of a brief PowerPoint presentation, but you will spend most of the time operating your KBE application. In this part you will also take care of guiding the assessment team through your Deliverable 2 report.

2. The **second part** (about 20-30 minutes) consists of a **Q&A session**, where you will keep operating the KBE tool under specific requests of the assessment team and you will answer questions related to your tool and/or delivered documentation. During this part you will also be invited to show some of your code and expose details of your implementation.

IMPORTANT NOTES:

- The live demonstration will imply the **physical presence of both team members**. No Skype or Webex sessions are possible.
- You will make use of your **own laptop(s)**. Beamer and projection screen will be provided. Make sure to come with proper installation of software, valid licenses, etc. Do not forget mouse, power adapter and, in case you have a MAC laptop, **video cable** adapter.
- You will bring **two printout copies** of Deliverable 2.
- You KBE application and related documentation must be present and ready to download from BB right before the beginning of your own live demo. The live demonstration will start with you **downloading your KBE application** directly from Blackboard. It is not allowed to demonstrate a different application than the one uploaded on BB.
- It is mandatory that **both team members** demonstrate their ability to operate the KBE application.
- Make sure to have a **CAD system** and **Q3D installed** on your laptop to demonstrate the specifically required functionalities of your KBE app
- It is very important that you **prepare this live demonstration well**. Make sure you have a “mental script” of what you want to show. Make sure all the functionalities of the application you want to demonstrate work properly. Time lost in last minute debugging and bad improvisation might prevent you to properly demonstrate the tool within the given and fixed time slot. **Any functionality that cannot be demonstrated during the live session will be assumed to be not present or not working!**
- You will receive your grade right after your live demo session

[3] In this case a jury composed by the course teachers and/or other FPP KBE specialists. You will find the detailed assessment criteria for the live demo posted on BB

Provided reference material

As for the development of any KBE application, a **knowledge acquisition phase** is required before starting with the generation of any code. In this phase you acquire the knowledge to capture into your KBE application. To this purpose you will need to learn the basics of aircraft tail design, in order to identify the various design steps, and collect all the necessary data and rules to be finally implemented in your KBE application.

To this purpose you are advised to make use of the Support Material Package, which you will find **posted on BB** next to this assignment.

This package consists of a selected set of slides including data, sizing and analysis methods relevant to aircraft and tail design:

- Heuristic method for preliminary longitudinal position of the wing
- Graphical method for MAC determination
- Tail volume coefficient method for horizontal and vertical tailplane sizing (definition plus statistical data)
- Tail shape sizing data (aspect ratio, sweep, thickness, etc.)
- Class II weight estimation for tailplane component (Raymer's method)
- Tailplanes staggering method to avoid spin
- Horizontal tail-wing relative position to avoid pitch up and deep stall problem
- Wing Downwash gradient estimation method
- Estimation methods for aerodynamic center contributions
- Estimation method to determine the most aft c.g. position complying with stability requirements
- DATCOM method for lift gradient estimation of lifting surfaces
- Sizing method for V tails

Final notes & Tips & Recommendations:

- The tool will be developed using the GDL KBE system.
- The UML diagrams are supposed to help you twice:

    1. **Before** you start developing any code. You can use these diagrams to "design" your KBE application before "manufacturing" it.

    2. **After** you developed your application, as a form of documentation. The UML diagrams are supposed to enable your colleagues and customer to understand the structure of your application without going to the source code.

- Find on BB all the necessary information on UML diagrams (see folder *on UML*)

- To build your UML diagrams you can use Microsoft Visio. The basic version of Visio offers all the necessary UML shapes and connectors, although you might find convenient to install this external UML package (http://www.softwarestencils.com/uml/index.html) which allows a faster generation of less cluttered plots. Several free UML editors can be found on the web. In any case, do not re-invent UML!!!

- For any calculation requiring geometry data as input (such as areas, lengths, distances, etc.) make sure to extract all the possible geometrical information directly from the GDL model. Hence, do not estimate areas, volumes, etc. using heuristics or analytical approximations when you can extract that info directly from the instantiated geometrical model!

- **Never hard code** any numbers inside your application! Always expose all the used values. Use external editable files when possible. The use of **more input files** is recommended to separate the input data that need to be edited by the user to define the design case at hand, from those that are more invariant and have the role of general setting parameters.

- Your KBE application should be easy to debug and modify/expand, so make sure to **extensively comment your code** and organize it in a modular way. Make use of **multiple .lisp files**, properly stored in separated directories when opportune. Preferably, use a separate lisp file for each *define-object* and *defun*

- Create a *readme* file with the basic instructions to allow others operating your application

- Test your application for **significantly different design cases** to check the robustness and flexibility of your application.