



D1.7 Standardization recommendations for data and model databases and tools

Yasser Meddaikar, Thimo Kier, Matthias Wuestenhagen (DLR),
Fanglin Yu (TUM), Charles Poussot-Vassal (ONERA), Bela
Takarics, Balint Vanek (SZTAKI)

GA number: 815058
Project acronym: FLIPASED
Project title: FLIGHT PHASE ADAPTIVE AEROSERVO-
ELASTIC AIRCRAFT DESIGN METHODS
Funding Scheme: H2020 **ID:** MG-3-1-2018
Latest version of Annex I: 1.1 released on 12/04/2019
Start date of project: 01/09/2019 **Duration:** 40 Months

Lead Beneficiary for this deliverable:	DLR
Last modified: 21/06/2023	Status: Delivered
Due date: 30/11/2020	

Project co-ordinator name and organisation: Bálint Vanek, SZTAKI
Tel. and email: +36 1 279 6113 vanek@sztaki.hu
Project website: www.flipased.eu

Dissemination Level		
PU	Public	
CO	Confidential, only for members of the consortium (including the Commission Services)	X

"This document is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815058."

Glossary

ASE	Aeroservoelastic
AFS	Active Flutter Suppression
AVL	Athena Vortex Lattice
CAD	Computer-aided Design
CG	Centre of gravity
CPACS	Common Parametric Aircraft Configuration Schema
DLM	Doublet Lattice Method
FE	Finite Element
FSM	Force Summation Method
GLA	Gust Load Alleviation
LTI	Linear Time-invariant
MDO	Multidisciplinary Design Optimization
MLA	Manoeuvre Load Alleviation
MPC	Model Predictive Control
RCE	Remote Component Environment
SFC	Specific Fuel Consumption
VLM	Vortex Lattice Method
WRBM	Wing Root Bending Moment

Table of contents

1	Executive Summary	5
2	CPACS	6
2.1	Initial CPACS dataset - Demonstrator workflow	6
2.2	Initial CPACS dataset - Scale-up workflow	7
3	RCE	9
4	Models and interfaces in MDO demonstrator workflow	10
4.1	Aircraft model generation	10
4.2	NASTRAN aeroelastic model integration	12
4.3	ASE model integration	13
4.4	Flutter controller design	15
5	Models and interfaces in scale-up D150 workflow	17
5.1	CPACS dataset preparation	17
5.2	Aircraft loads analysis and design	18
5.3	ASE model integration	18
5.4	Loads Analysis	19
5.5	GLA controller design	20
5.6	Flutter controller design	20
5.7	Induced drag evaluation	22
5.7.1	Trefftz plane implementation in NASTRAN	22
5.7.2	VLM-based near-field implementation	22
5.7.3	PANUKL-based drag estimation tool	22
5.8	Aircraft mission evaluation	23
6	Results from RCE workflow	24
6.1	Design study with flutter mass and sweep angle variation	24
7	Conclusion and Outlook	25
8	Bibliography	26

List of Figures

1	CPACS dataset of demonstrator workflow	7
2	Outer geometry of the FLIPASED DLR-D150 configuration generated from the CPACS dataset using TiGL Viewer 2.1.3	8
3	RCE tools in the NASTRAN aeroelastic model integration block	10
4	RCE tools in the Aircraft model generation sec	11
5	RCE tools in the NASTRAN aeroelastic model integration block	12
6	Output files and format from NASTRAN aeroelastic model integration tool	13
7	RCE tools in the ASE model integration block	13
8	RCE workflow for the aeroelastic model reduction for flutter suppression control design .	15
9	D150 initial control surfaces	17
10	D150 modified control surfaces	17
11	1-cosine gust and aircraft gust zones.	19
12	MPC principle [12].	21
13	Reference flexible aircraft model defined by the structural grid (red), the aerodynamic panel model (blue), the deployed control surfaces for GLA (magenta) and the sensor coordinate system locations and orientations (black).	21

1 Executive Summary

The deliverable “D1.7 Standardization recommendations for data and model databases and tools” summarizes the interfaces and data formats that have been followed in the multidisciplinary design optimization (MDO) tasks within WP2 and WP4, that is, for the demonstrator and scale-up workflow respectively. The interfaces have been established over the course of the project enabling as automated of a dataflow as possible between the tools of different partners.

Three aspects influence the developed interface in the project - i) the use of CPACS as the aircraft definition norm, ii) RCE as the execution environment, and iii) the tools being used and their required inputs and outputs.

This document explains the currently existing interfaces for the two workflows. A brief introduction to CPACS and RCE are presented in the first chapters. This is followed by a description of the individual blocks incorporated in the workflow and the input and output data to each of the corresponding tools. Lastly, a selection of results as an example from the demonstrator workflow is presented.

The deliverable has been jointly created by all partners with contributions from DLR, SZTAKI, TUM and ONERA.

2 CPACS

CPACS (Common Parametric Aircraft Configuration Schema) is an open standard for exchanging and sharing aircraft design data. It was developed by the German Aerospace Center (DLR) in collaboration with industry partners and is maintained by the CPACS Initiative, a non-profit organization dedicated to promoting the use of CPACS.

CPACS defines a standardized format for describing the geometric, structural, aerodynamic, and systems properties of an aircraft. This includes information such as the wing and fuselage geometry, materials used, engine performance, and control surfaces. By using a common format for this information, designers, engineers, and researchers can more easily share and collaborate on aircraft design projects.

One of the key benefits of CPACS is its flexibility. It can be used to describe a wide range of aircraft configurations, from small drones to commercial airliners. Additionally, it can be customized to include specific design parameters and properties, allowing it to be tailored to the needs of different projects and organizations.

CPACS has been widely adopted in the aerospace industry, with major aircraft manufacturers, research organizations, and universities using it for their design and analysis activities. It has also been integrated into a number of commercial software tools, such as computer-aided design (CAD) software, aerodynamic analysis software, and optimization tools.

The use of CPACS has several advantages. It allows for more efficient collaboration and communication among designers, engineers, and researchers. It also enables faster design iteration and optimization, as the data can be easily exchanged and analyzed. Additionally, it facilitates the development of automated design tools and workflows, which can lead to significant time and cost savings.

2.1 Initial CPACS dataset - Demonstrator workflow

The CPACS dataset of the demonstrator workflow includes the following information:

- the geometry information for the wing, fuselage, and V-tails
- the structure definitions for the wing and V-tails
- airfoil data

User-defined tool-specific information is stored in the `toolspecific` field of the CPACS dataset, allowing customization without compromising the default data format and maintaining the flexibility of the dataset. As an example, in the case of drag reduction, the software AVL or Athena Vortex Lattice is utilized, and specific setups for AVL are defined within this field.

The design study fields in CPACS are utilized to facilitate the parameter study of the demonstrator workflow. In these fields, the parameters to be investigated are defined, providing documentation for each run of the parameter study. This enables the analysis and exploration of various parameter configurations within the workflow.

The CPACS dataset is generated using a Python script that leverages the TIXI library. The TIXI library, developed by DLR, is specifically designed to support the handling of CPACS data. This library plays a crucial role in facilitating the generation and manipulation of CPACS datasets within the Python script. The figure 1 shows the generated CPACS dataset.

Node	Content
xml	version="1.0" encoding="utf-8"
cpacs	all(header, vehicles?, missionDefinitions?, airports?, flights?, airlines?, studies?, toolsspecific?)
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation	cpacs_schema.xsd
header	all(name, description?, creator, timestamp, version, cpacsVersion?, updates?)
vehicles	all(aircraft?, rotorcraft?, engines?, profiles?, structuralElements?, materials?, fuels?)
materials	(material+, composites?)
aircraft	(model+)
model	all(name, description?, reference?, fuselages?, wings?, engines?, enginePylons?, landingGear?, systems?, genericGeometryComponents?, global?, analyse...)
uid	FLEXOP
name	FLEXOP
reference	all(area?, length?, point?)
fuselages	(fuselage+)
wings	(wing+)
wing	all(name, description?, parentUID?, transformation, sections, positionings?, segments, componentSegments?, dynamicAircraftModel?)
uid	WR
symmetry	x-z-plane
name	WR
description	WR
parentUID	FU
transformation	all(scaling?, rotation?, translation?)
sections	(section+)
positionings	(positioning+)
segments	(segment+)
componentSegments	(componentSegment+)
componentSegment	all(name, description?, fromElementUID, toElementUID, structure?, controlSurfaces?, path?, wingFuselageAttachments?, wingWingAttachments?, wingFu...)
uid	comSeg_WR
name	cs_WR
description	cs_WR
fromElementUID	ele_sec_WRbase
toElementUID	ele_sec_WRtip
structure	all(upperShell, lowerShell, intermediateStructure?, ribsDefinitions?, spars?)
controlSurfaces	all(leadingEdgeDevices?, trailingEdgeDevices?, spoilers?)
wing	all(name, description?, parentUID?, transformation, sections, positionings?, segments, componentSegments?, dynamicAircraftModel?)
analyses	all(aeroPerformance?, aeroelastics?, dynamicAircraftModel?, flightDynamics?, flyingQualities?, flightPerformance?, flightSystems?, landingGearPositionS...)
profiles	all(fuselageProfiles?, wingAirfoils?, guideCurves?, rotorAirfoils?, structuralProfiles?, nacelleProfiles?, curveProfiles?)
wingAirfoils	(wingAirfoil+)
fuselageProfiles	(fuselageProfile+)
studies	all(parameters, designStudies)
parameters	(parameter+)
parameter	all(name)
designStudies	(designSpace+)
designSpace	all(designParameters, stateParameters, status)
toolspecific	(tool+)
avl	

Figure 1: CPACS dataset of demonstrator workflow

2.2 Initial CPACS dataset - Scale-up workflow

Within the frame of FLIPASED, the need for a reference model for the scale-up task in WP4 was recognized. This reference model would be used as the baseline for the multi-disciplinary optimization (MDO) tasks in WP4. The DLR-D150 was chosen as the reference configuration and its CPACS dataset was provided by DLR Institute of Aeroelasticity (DLR-AE).

The CPACS dataset includes the CPACS definition (CPACS Version 2.3) of the DLR-D150 configuration (geometry, definition of the structure for the wing and tail, mass model data, material), adapted from its status in the ILOADS project.

In the FLIPASED DLR-D150 CPACS dataset, composite materials have been used for the wing. The CPACS dataset has also been adapted to be used as input file for the aeroelastic design process cpacs-MONA [11] at DLR-AE.

An illustration of the aircraft outer geometry corresponding to this CPACS dataset using TiGL Viewer 2.1.3 is shown in Figure 2.

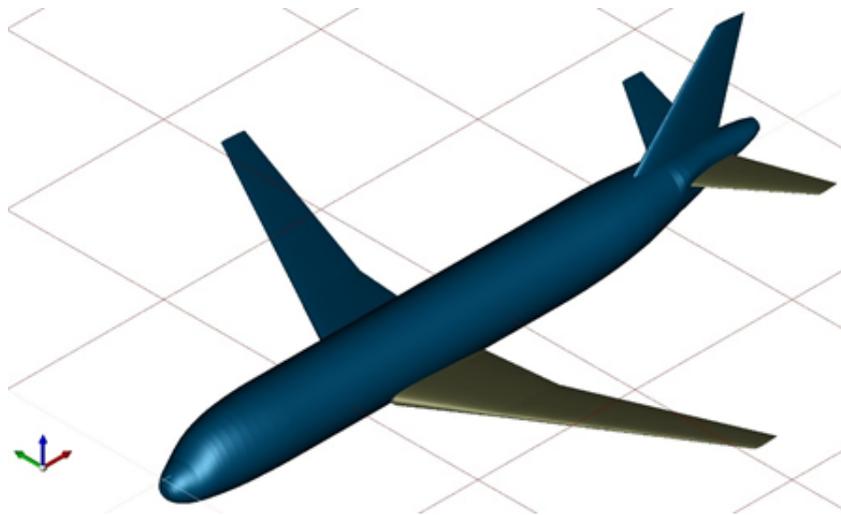


Figure 2: Outer geometry of the FLIPASED DLR-D150 configuration generated from the CPACS dataset using TiGL Viewer 2.1.3

3 RCE

RCE (Remote Component Environment) is an open-source software framework for building and executing scientific workflows and applications. It was developed by the Helmholtz Center for Environmental Research in Germany, and is now maintained by an international community of developers.

RCE provides a graphical user interface for designing and executing workflows, which are composed of individual components that perform specific tasks. Components can be written in a variety of programming languages, and can be executed locally or on remote systems. RCE also supports parallel execution of components, which can improve performance and reduce processing times.

One of the key features of RCE is its ability to integrate with a wide range of scientific software and tools. This includes software for data analysis, simulation, visualization, and more. RCE provides a standardized interface for interacting with these tools, making it easier to incorporate them into scientific workflows.

RCE also includes a number of features for managing data, including versioning, access control, and replication. This makes it easier to collaborate on scientific projects, share data and workflows, and maintain data integrity and consistency.

RCE is widely used in the scientific community for a variety of applications, including environmental modeling, bioinformatics, and computational fluid dynamics. Its flexibility, scalability, and integration capabilities make it a valuable tool for researchers and scientists who need to process large amounts of data and perform complex analyses.

4 Models and interfaces in MDO demonstrator workflow

The demonstrator workflow is established to performed design studies on variants of the existing flutter wing. The studies aim at varying parameters such as the aspect ratio and wing sweep, while optimizing the demonstrability of active flutter suppression (AFS) (the difference between closed loop and open loop flutter speeds). The workflow is set up in the RCE environment with locally hosted tools at each partner. The data exchange between the different tools which forms the basis of the model and data interfaces is described in this chapter.

A schematic of the entire workflow is shown in Figure 3.

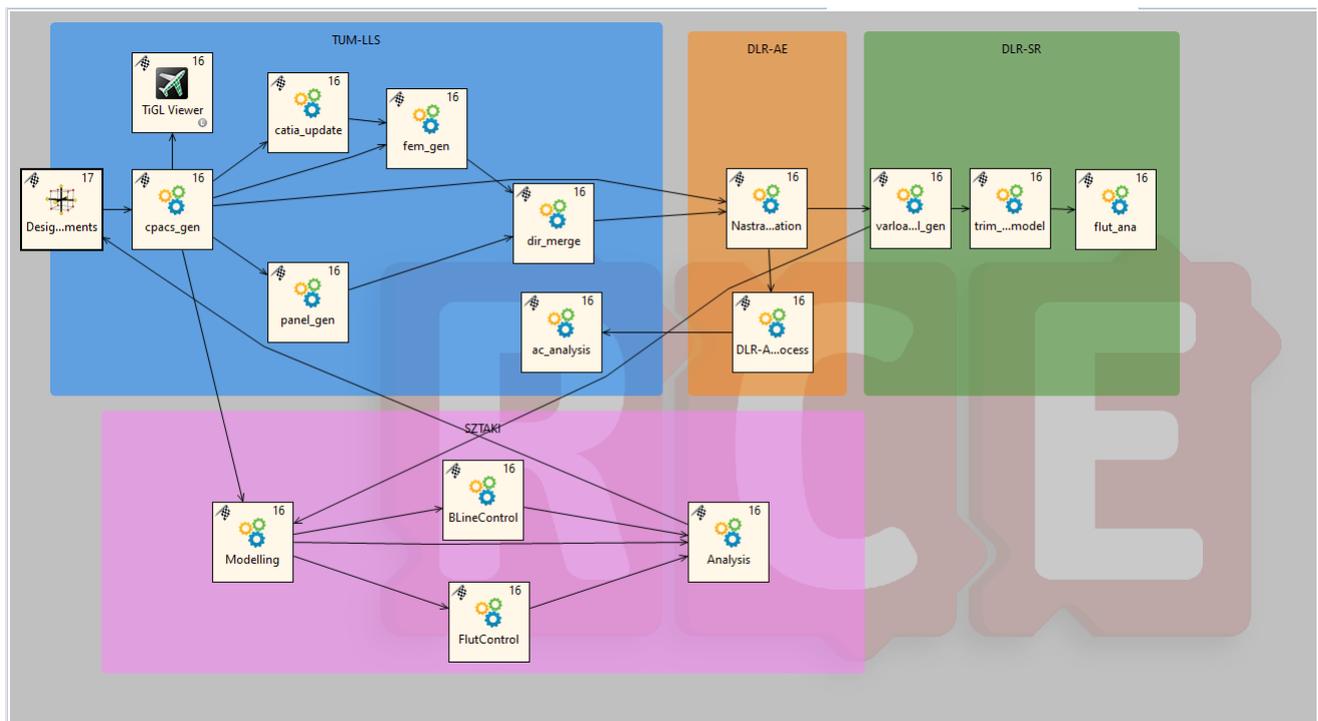


Figure 3: RCE tools in the NASTRAN aeroelastic model integration block

4.1 Aircraft model generation

The Aircraft Model Generation section of the demonstrator workflow consists of several blocks and functions. The CATIA block updates the geometry and structure of the demonstrator using the CPACS dataset. The Hypermesh block generates the finite element (FE) model based on the updated CATIA model. The Aero model block generates the Doublet Lattice Method (DLM) model for NASTRAN based on the geometry information from CPACS. To optimize the model generation process, the FE model and DLM generation processes are parallelized. Finally, the two models are merged together and passed on to the next section of the demonstrator workflow. Figure 4 illustrates the connection between the mentioned tools.

In terms of the data-flow to each tool, the following Table 1 summarize the inputs and outputs to and from each of the tools.

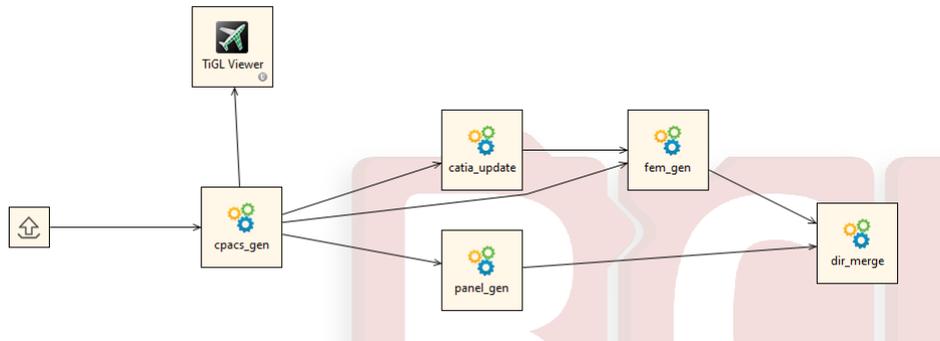


Figure 4: RCE tools in the Aircraft model generation sec

Table 1: Tools, inputs and outputs in the aircraft model generation section

Tools	Inputs	Outputs
CATIA update	<ul style="list-style-type: none"> CPACS dataset 	<ul style="list-style-type: none"> Folder containing updated CATIA model
FE model generation	<ul style="list-style-type: none"> CPACS dataset Folder containing updated CATIA model 	<ul style="list-style-type: none"> Folder containing containing bulk data files corresponding to the structure of wing
Panel generation	<ul style="list-style-type: none"> CPACS dataset 	<ul style="list-style-type: none"> Folder containing containing bulk data files corresponding to the DLM of wing
Directory merge	<ul style="list-style-type: none"> Folder containing containing bulk data files corresponding to the structure of wing Folder containing containing bulk data files corresponding to the DLM of wing 	<ul style="list-style-type: none"> Folder containing containing bulk data files corresponding to the wing

4.2 NASTRAN aeroelastic model integration

The NASTRAN aeroelastic model integration section of the demonstrator workflow serves two purposes.

- To assemble the full aircraft aeroelastic model in NASTRAN and generate as outputs, system matrices and other files required for assembling the MATLAB aeroservoelastic (ASE) model at DLR-SR.
- To execute post-processing tools performing trim and flutter analyses and to pass these NASTRAN decks to other partners.

In terms of the data-flow to each tool, the following Table 2 and Figures 5-6 summarize the inputs and outputs to and from each of the tools.

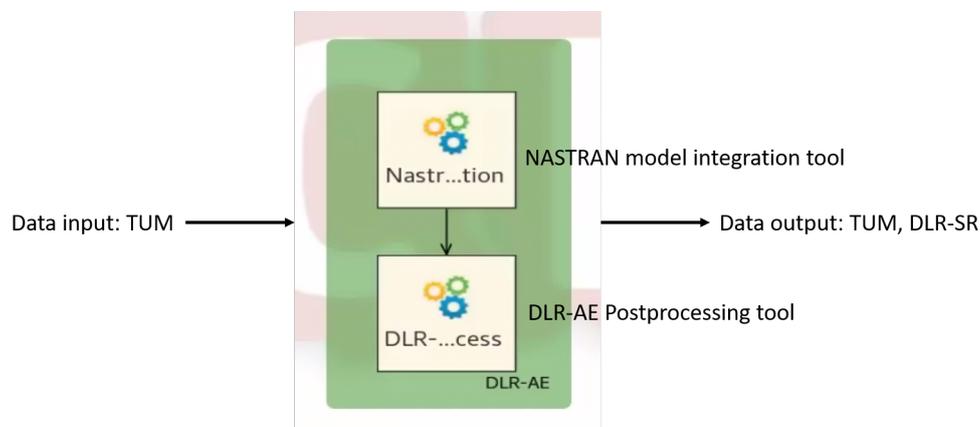


Figure 5: RCE tools in the NASTRAN aeroelastic model integration block

Table 2: Tools, inputs and outputs in the NASTRAN aeroelastic model integration block

Tools	Inputs	Outputs
NASTRAN aeroelastic model integration	<ul style="list-style-type: none"> • CPACS dataset • Folder containing bulk data files corresponding to wing 	<ul style="list-style-type: none"> • Folder containing model data - system matrices (stiffness, mass), aerodynamic panel model definition, outputs defining the condensation grids (summarized in Figure 6) • CPACS dataset
DLR-AE post-processing		<ul style="list-style-type: none"> • NASTRAN solution decks for modal, aeroelastic trim and flutter analyses

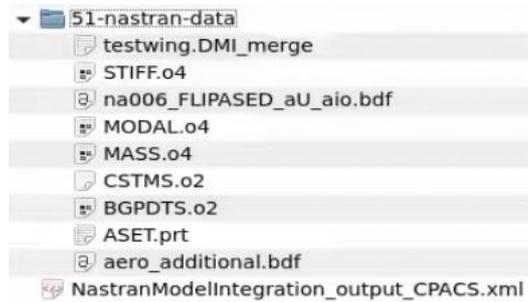


Figure 6: Output files and format from NASTRAN aeroelastic model integration tool

The outputs from NASTRAN are controlled using appropriate DMAP alters and include both binary text based data formats. In terms of the aerodynamic panel model, the panel definitions, control surface definitions and the aggregated W2GJ correction due to camber and twist are included.

4.3 ASE model integration

The ASE model integration section shown in Figure 7 has the following tasks:

- assemble the data provided by the NASTRAN aeroelastic model integration to an ASE model. The results of this intermediate step are also passed on to the flutter controller design.
- trim the created model at different flight conditions of interest and create linearized models.
- analyze based on the linearized models at what speed and frequency flutter becomes unstable

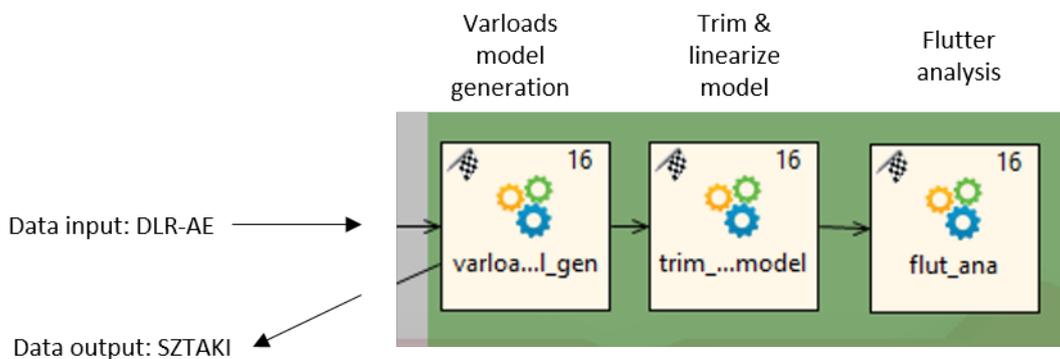


Figure 7: RCE tools in the ASE model integration block

Table 3 summarizes how the connection of the ASE model integration part within the entire workflow is established including information on what data is received and how it is processed and passed on.

As the tools of the ASE model integration part are all hosted in Matlab, the corresponding outputs and results are saved as mat-files.

Table 3: Tools, inputs and outputs in the ASE model integration block

Tools	Inputs	Outputs
Varloads model generation	<ul style="list-style-type: none"> Folder containing model data - system matrices (stiffness, mass), aerodynamic panel model definition, outputs defining the condensation grids (summarized in Figure 6) CPACS dataset 	<ul style="list-style-type: none"> directory with ASE data set, in a format it can be read by the Matlab tools used by DLR and SZTAKI, containing information on the structural dynamics and aerodynamics, as well as actuator, engine and sensor dynamics
trim & linearize model	<ul style="list-style-type: none"> directory with ASE data set, in a format it can be read by the Matlab tools used by DLR and SZTAKI, containing information on the structural dynamics and aerodynamics, as well as actuator, engine and sensor dynamics 	<ul style="list-style-type: none"> directory with trimming results and corresponding linearized models
flutter analysis	<ul style="list-style-type: none"> directory with trimming results and corresponding linearized models 	<ul style="list-style-type: none"> directory with flutter analysis results

4.4 Flutter controller design

The model order reduction and flutter control design blocks shown in Figure 8 have the following tasks:

- Determine if the linear time-invariant (LTI) ASE model delivered by the ASE model integration has unstable flutter modes.
- Reduce the LTI model order for flutter suppression control design.
- Design the flutter suppression controller.
- Analyze the flutter controller.

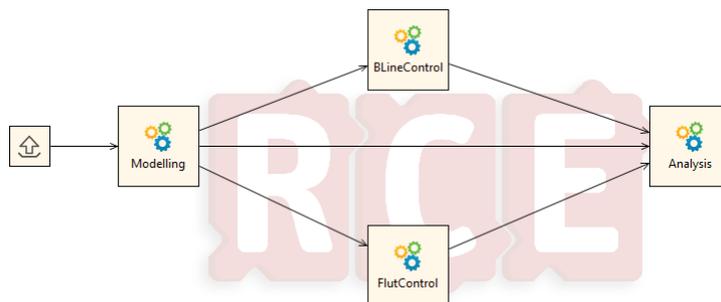


Figure 8: RCE workflow for the aeroelastic model reduction for flutter suppression control design

Table 4 provides a summary of how the flutter suppression control design block is connected with the previous blocks of the workflow, what data is received and how it is processed and passed on. The baseline control design block is not used at this stage.

Table 4: Tools, inputs and outputs in the flutter control design block

Tools	Inputs	Outputs
Modeling	<ul style="list-style-type: none"> • Folder containing LTI ASE model • CPACS dataset 	<ul style="list-style-type: none"> • Folder containing the reduced order LTI ASE model • CPACS dataset
Flutter control design	<ul style="list-style-type: none"> • Folder containing the reduced order LTI ASE model • CPACS dataset 	<ul style="list-style-type: none"> • directory with the resulting flutter suppression controller given in LTI structure • CPACS dataset
Flutter controller analysis	<ul style="list-style-type: none"> • Folder containing the reduced order LTI ASE model • directory with the resulting flutter suppression controller given in LTI structure • CPACS dataset 	<ul style="list-style-type: none"> • directory with flutter analysis results • CPACS dataset

5 Models and interfaces in scale-up D150 workflow

In the scale-up task within WP4, the DLR-D150 is used as a reference model. In the present implementation, the data exchange between different partners is executed outside of RCE. The workflow can be set up within RCE similar to the demonstrator workflow in a next step.

The primary goal of the study is to observe trends with varying aspect ratio - aircraft weight, open-loop loads, closed-loop loads with manoeuvre load alleviation (MLA) and gust load alleviation (GLA), aircraft critical flutter speed (open-loop and with ASF), and fuel burn (open-loop and with active wing-shape control for minimum induced drag). The study is meant to illustrate the performance benefits attainable through different active controls technologies. The present chapter summarizes the data interfaces between the different tools in this MDO task.

5.1 CPACS dataset preparation

The initial CPACS dataset of D150 includes one inner flap, one outer flap, and one aileron, which matches the control surface allocation of the A320. However, in order to fully harness the potential of MLA, GLA, and wing shape control, the consortium has decided to increase the number of control surfaces. The provided figure 10 displays the modified CPACS dataset, which now consists of 2 inner flaps and 8 outer control surfaces. Depending on the design requirements, these 8 outer control surfaces can be grouped together as outer flaps or ailerons, providing the necessary flexibility for MLA, GLA, and wing shape control.

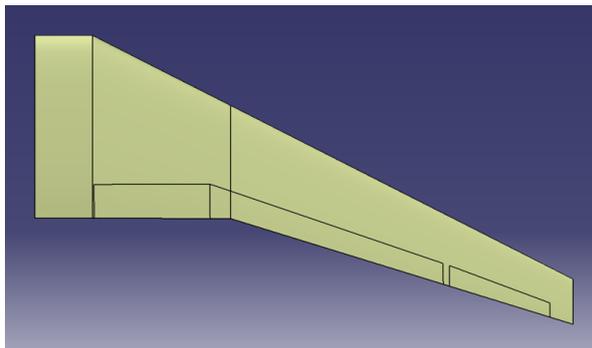


Figure 9: D150 initial control surfaces

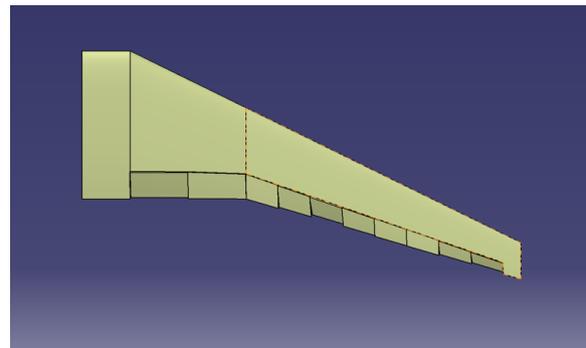


Figure 10: D150 modified control surfaces

As depicted in the figure 9, the initial D150 CPACS dataset displayed the outer flap penetrating into the kink area where the inner flap is located. This configuration was deemed unrealistic and introduced additional modeling complexities. Therefore, it has been rectified to eliminate such penetration. Furthermore, considering the current design stage, it was deemed unnecessary to have a separation between the outer flap and aileron, as it complicated the modeling process. Hence, this separation will be removed for the sake of simplification.

As previously mentioned, the main objective is to examine trends in relation to varying aspect ratios. Due to time constraints, the range of aspect ratios has been limited to 9.4 and 18.4. The initial CPACS dataset has an aspect ratio of 9.4. The value of 18.4 is chosen as a reasonably high value that may be achievable with improved technology in the future. In total, six different configurations will be investigated to demonstrate a meaningful trend in relation to aspect ratio.

To prevent the design space from becoming excessively large and to control the influencing factors,

certain preconditions have been defined. One of these preconditions is to maintain a similar wing planform by keeping the sweep angle and taper ratio constant. Additionally, the wing area is kept constant to ensure consistent wing loading, thereby avoiding any significant impact on aerodynamic performance and load analysis of the wing. By imposing these preconditions, the focus can be directed towards investigating the influence of varying aspect ratios while minimizing other potential variables.

In order to vary the aspect ratio, adjustments are made to the wing's span. The first segment of the wing, which lies within the fuselage, maintains a constant length since the fuselage itself remains unchanged. Therefore, only the segment of the wing outside of the fuselage is scaled according to the desired aspect ratio. To ensure the wing area remains constant, the root chord is adjusted as the wing span is modified. This approach allows for controlled changes in aspect ratio while maintaining a consistent wing area and fuselage dimensions.

As the aspect ratio increases, the wing span is extended accordingly and the wing becomes more slender. This change, along with the sweep angle, causes the aerodynamic center to shift further towards the rear. Consequently, it can affect the flight dynamics of the aircraft. To ensure that the aircraft remains stable and to facilitate the design of a baseline controller, a stability check is conducted using SUAVE [3], an aircraft design tool developed by Stanford University.

During this stability check, the wing position is adjusted to maintain a constant neutral point. By keeping the neutral point consistent, the stability characteristics of the aircraft can be preserved despite the changes in aspect ratio and wing configuration. This process helps ensure that the aircraft remains controllable and stable throughout its flight envelope.

5.2 Aircraft loads analysis and design

The in-house tool cpacs-MONA at DLR-AE is used for the structural design of the aircraft together with a comprehensive aircraft loads process. A description of the cpacs-MONA process is presented in [11].

The input to cpacs-MONA is a CPACS dataset with the aircraft definition. This corresponds to the dataset of the DLR-D150 shared with the consortium, with modifications made to reflect the changed aspect ratio and number of control surfaces on the wing.

The output from the cpacs-MONA tool can be summarized as the following.

- Stiffness matrix corresponding to the condensed aircraft model (after structural optimization)
- Mass cards corresponding to different mass and center of gravity (CG) configurations
- DLM aerodynamic model in NASTRAN
- Definition of loadcases considered and the down-selected loadcases used in the structural optimization

5.3 ASE model integration

During the ASE model integration two main steps happen:

1. NASTRAN decks are received from cpacs-MONA and the aeroelastic data is generated for a the Simulink model representing a flexible aircraft

2. The Simulink model is trimmed and linearised for the load cases which are cpacs-MONA found to be the most critical ones

The set of linearised models is important for the simulations performed during the loads analysis and for the synthesis of the MLA and GLA control.

5.4 Loads Analysis

A loads analysis is performed based on the linearised models which in a first step provides the open-loop model behaviour for various gust encounters. At this point it can be validated if the critical loads calculated match with the ones defined by cpacs-MONA. The activity of the primary flight control, MLA and flutter controller are neglected. Fundamentally, however, all control law functions affect the loads. As soon as a GLA controller is synthesised the loads analysis can be performed in closed-loop. The worst case loads are then fed back to the structural sizing performed by cpacs-MONA.

The performance of the GLA and MLA control is judged based on the loads P_c , which the wing structure experiences due to gust encounter. The bending moment $P_{c,max}$ is of special interest. The loads are estimated with the force summation method (FSM)

$$P_c = T_{cg} \left(P_g^{ext} - P_g^{iner} \right) \quad (1)$$

where the external and inertial loads are P_g^{ext} and P_g^{iner} . With matrix T_{cg} the incremental loads of the load monitoring points along the wing are summed up and transformed to the loads coordinate system from the wing tip up to the considered load monitoring position [2, 10].

Various vertical 1-cosine gust profiles serve as gust inputs, which are defined by the gust zone velocity and acceleration $U_{z,t}(t)$ and $\dot{U}_{z,t}(t)$

$$U_{z,t}(t) = \begin{cases} \frac{\bar{U}_t}{2} \left(1 - \cos \left(\frac{\pi}{H_t} (U_\infty t - x_z) \right) \right), & \text{if } \frac{x_z}{U_\infty} \leq t \leq \frac{2H_t + x_z}{U_\infty} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$\dot{U}_{z,t}(t) = \begin{cases} \frac{\bar{U}_t \pi}{2H_t} U_\infty \sin \left(\frac{\pi}{H_t} (U_\infty t - x_z) \right), & \text{if } \frac{x_z}{U_\infty} \leq t \leq \frac{2H_t + x_z}{U_\infty} \\ 0, & \text{otherwise.} \end{cases}$$

The maximum gust intensity and gust half length are \bar{U}_t and H_t [4]. With evolving time t the aircraft flies through the gust from nose to aft. This is shown in Figure 11. The aerodynamic model of the aircraft

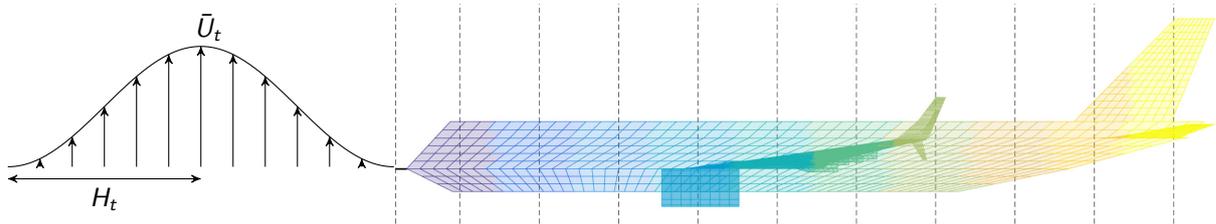


Figure 11: 1-cosine gust and aircraft gust zones.

is separated in gust zones as indicated by the different colours of the aerodynamic panel model. All panels belonging to the same gust zone are assumed to experience the same gust velocity observed at the centre line defined at position x_z . The gust zones are separated by the vertical dashed lines. Namely, within a gust zone the gust velocity is constant. The air data boom at the nose is treated as a gust zone by its own. Angle of attack α changes are recognised there first. For GLA control a feedforward path can be used [13]. The gust zone approach is an approximation. It saves computation time as it groups many aerodynamic panels. With ten gust zones the implementation was found to be quite accurate [8]. The gust velocity difference of two neighbouring zones is a time delay dependent on the airspeed U_∞ . As a transfer function a time delay can be defined by

$$G_{z,d}(s) = e^{-t_{z,d}s}, \quad (3)$$

where $t_{z,d}$ is the time delay in seconds and s is the Laplace variable [8]. A second-order Padé approximation of a time delay is

$$G_{z,d}(s) \approx \frac{s^2 - \frac{6}{t_{z,d}}s + \frac{12}{t_{z,d}^2}}{s^2 + \frac{6}{t_{z,d}}s + \frac{12}{t_{z,d}^2}}. \quad (4)$$

It converts to a linear state-space system [6]. Thus, the inputs to the gust zones reduces to the inputs $U_{z,g}$ and $\dot{U}_{z,g}$ at the air data boom. The gusts then propagate over all gust zones.

For the MLA the loads analysis is straight forward. The aircraft needs to be trimmed for the considered manoeuvre, while the load especially on the wing root is reduced. This can be done by shifting the required lift more inboards. Outboard control surfaces tend to be deflected upwards while the inboard ones are deflected downwards. Thus, the MLA control reduces to an optimised allocation of the control surfaces. The loads of interest can then also be analysed with the FSM.

5.5 GLA controller design

The GLA control is synthesized based on model predictive control (MPC). Figure 12 depicts the general principle of MPC. With MPC a system is controlled so that it follows a predefined trajectory. MPC predicts at the current time step k the output behaviour of a plant model n_p time steps into the future, where n_p is the prediction horizon. It optimises the input signals for the next n_c time steps to achieve the desired trajectory. The change in input is considered constant for time steps between $k + n_c$ and $k + n_p$ [1]. MPC then applies the first predicted control input increment. A time step of Δt_s later, the optimisation is repeated [12].

For GLA the elevators and ailerons on both wings are used. In Figure 13 the control surfaces are framed in magenta. Symmetric allocation of the control surfaces on the left and right side is performed for the considered vertical gust encounters. The GLA controller processes the α_a measurement at the air data boom, the z -accelerations and x -rotational rates taken from the fuselage IMU and the most inner and outer IMUs at the rear spar of each wing. The wing root bending moment (WRBM) $P_{c,mx}$ is estimated based on the given measurements.

5.6 Flutter controller design

The flutter suppression design block uses the inputs as described in the previous section. The first step of the flutter control design tool is first to evaluate whether the open loop model contains unstable flutter dynamics. If this condition holds, a flutter control needs to be synthesized based on the LTI model of

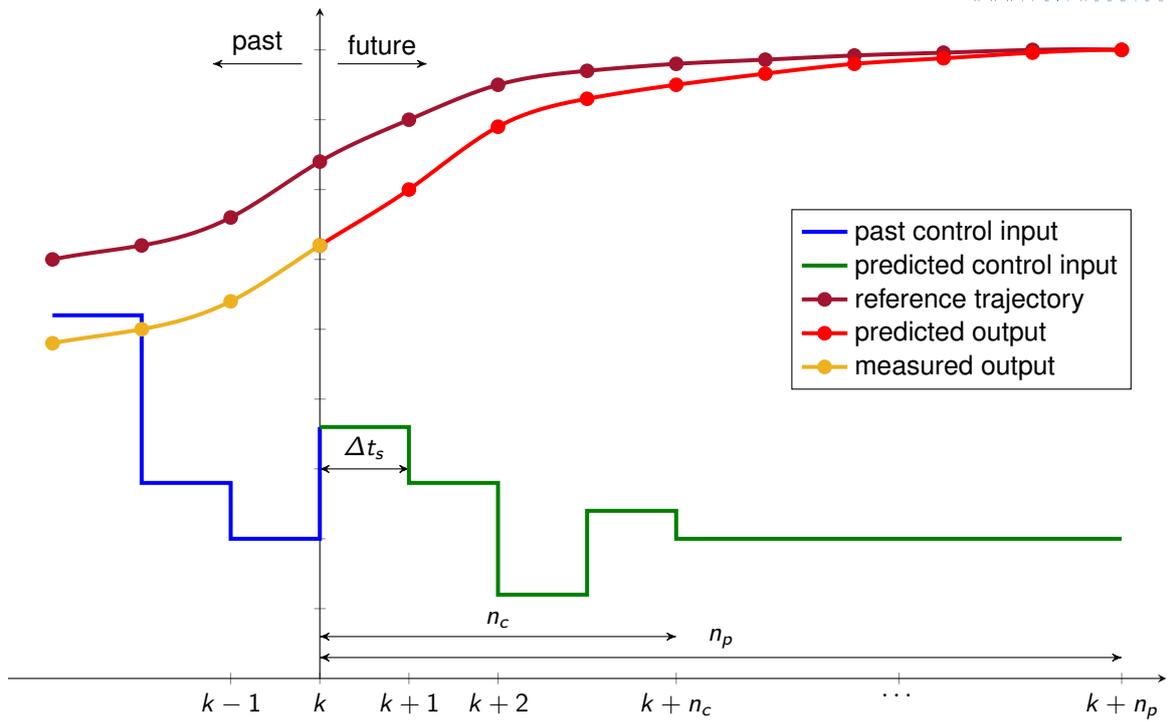


Figure 12: MPC principle [12].

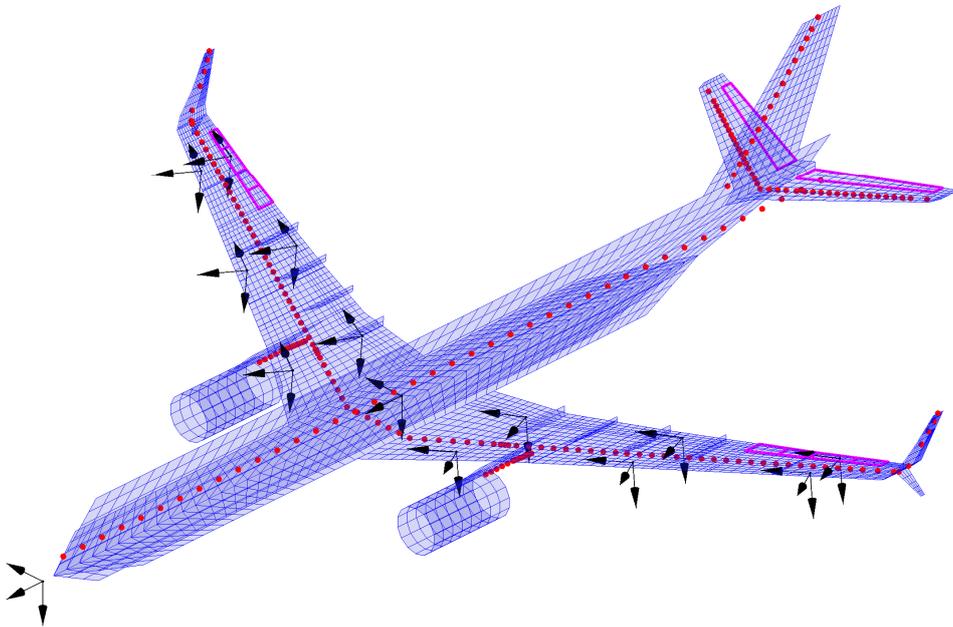


Figure 13: Reference flexible aircraft model defined by the structural grid (red), the aerodynamic panel model (blue), the deployed control surfaces for GLA (magenta) and the sensor coordinate system locations and orientations (black).

the open loop model. The LTI model of the aircraft is obtained via the ASE model integration block, which is then delivered to the flutter control design block in a compressed format accompanied by the corresponding CPACS file. The input of the flutter controller consists of the pitch rate (q), and angular rate measurement from the IMU sensors (q_L and q_R) placed along the wing. The actuating signals are the deflection commands for the pair of outermost ailerons. The controller is designed for the reduced order model with structured H_∞ synthesis. The state-space model of the resulting flutter suppression controller is the output of the block. The controller is saved in the ToolSpecific section of CPACS under the name Flutter.

The analysis of the closed-loop is based on disk margin calculations. Complex scalar uncertainties are injected into the channels involved in the feedback loops and the phase and gain combination at which the closed-loop becomes unstable is computed in each channel, simultaneously. The results of the flutter controller analysis block is the open loop flutter speed and the robust closed loop flutter speed.

5.7 Induced drag evaluation

For the induced drag modelling, three different tools were developed and tested. Their interfaces and utilized models are summarized below.

5.7.1 Trefftz plane implementation in NASTRAN

The Trefftz plane implementation is programmed within the SOL200 solution in MSC.NASTRAN, making use of appropriate cards to extract lift responses and to define the equations to compute the induced drag. The routine is coupled to an external Python script to perform the drag optimization. Several random distributions of the control surfaces are generated first, for each which the induced drag is calculated. The data points are used to construct a Kriging-Regression model, on which the minimization problem is solved.

The input to this tool includes: aircraft condensed or full FE model, the aircraft aerodynamic DLM model in NASTRAN, flight parameters for which the drag optimization is to be performed and deflection limits for the control surfaces.

The output from the tool is an Excel table containing the induced drag, aircraft trim variables, control surface deflections and span-wise lift values. This is at each of the control surface deflection combinations - the ones used to construct the regression model and for the optimal deflections obtained from the surrogate.

5.7.2 VLM-based near-field implementation

A vortex lattice method (VLM) - based near-field implementation [9] was studied as a candidate tool in this exercise. The work presented in [9] extends the classical VLM implementation in the loads environment VarLoads [7] in MATLAB to also include induced drag by accounting for in-plane forces. The optimization of the control surface allocation is performed in this case in MATLAB using the *fmincon* routine.

The inputs and outputs from the tool are similar to those in the NASTRAN-based tool described earlier in Section 5.7.1.

5.7.3 PANUKL-based drag estimation tool

PANUKL is a software package to compute the aerodynamic characteristics of an aircraft using low order panel methods [5]. The PANUKL framework consists of several programs, four of which are used in this investigation. The four programs, in logical order are listed below.

- Mesh3: Generates the investigated geometry mesh.
- Neigh: Calculates the connections of the generated panel mesh elements.
- Panukl: Performs the aerodynamic calculations.
- Press: Defines the important variables (lift force, pitching moment, etc.)

To achieve true trim flight conditions, the elastic deformation of the flexible structure needs to be taken into account. In this case, surface spline theory is used, which enables the transformation of aerodynamic forces and moments to the structural model and structural deformation to the aerodynamic model. The result is an iterative process with the undeformed aircraft geometry and structural properties as the input and the deformed geometry as the output.

The input to this tool includes: aircraft condensed FE model and the spine grid geometry data. The outputs from the tool are similar to those in the NASTRAN-based tool described earlier in Section 5.7.1.

5.8 Aircraft mission evaluation

The fuel requirements for different segments of a flight, such as taxi, takeoff, climb, descent, approach, landing, and contingency, are assumed to be constant, including an additional reserve of fuel. Therefore, only the cruise segment needs to be considered for evaluating fuel consumption. To simplify the analysis, the cruise segment is divided into smaller parts with consistent mass properties. Each step in the cruise segment requires a model of the D150 aircraft with corresponding mass properties to be created in order to estimate fuel consumption.

The optimal altitude for the cruise segment is determined based on the aircraft's polar, which corresponds to flying at the maximum lift-to-drag (L/D) ratio. This polar is derived from calculations of induced drag, accounting for some assumed parasitic drag components. Additionally, controlling the shape of the wings reduces drag even further.

The engine used for the aircraft is selected in advance and remains unchanged throughout the design workflow. Therefore, the engine characteristics are known, including a typical specific fuel consumption (SFC) value that can be assumed.

The primary criterion evaluated for the mission is the range achieved during the cruise segment. To analyze this, different fuel states along a defueling vector in the CG diagram need to be prepared. For each fuel state, the flexible aircraft is trimmed at a specified starting flight point. By considering the required thrust and the SFC of the engine, the flight time to reach the next fuel state is calculated. This flight time, along with the velocity, determines the range of the segment. At specific fuel states, a step climb is initiated to adjust the altitude according to the current aircraft mass while maintaining the optimal lift coefficient (C_L). The sum of all the ranges between the different mass states represents the objective function that needs to be maximized.

It is assumed that the use of AFS, GLA, MLA, and wingshape control can further enhance the range capabilities of the aircraft.

The inputs to the aircraft mission evaluation block include the CPACS dataset, the estimated C_L and C_D at the different cruise segments, and the aircraft mass at the start and end of each cruise segment. The output of the block is the calculated flight range for the given configuration.

6 Results from RCE workflow

In this chapter, a selection of the results obtained from the demonstrator workflow described in Chapter 4 is presented. The results are from the automated workflow established within RCE.

6.1 Design study with flutter mass and sweep angle variation

The following section describes the results of the demonstrator RCE workflow. In this workflow the flutter mass and the sweep angles were varied as presented in Table 5. The final results of the workflow provide the open loop flutter speed of the aircraft, the robust closed loop flutter speed of the aircraft and the possible increase in the flutter speed with active control. The results are shown in Table 5.

Table 5: Demonstrator RCE results

Flutter mass [kg]	Sweep angle [deg]	Open loop flutter speed [m/s]	Closed loop robust flutter speed [m/s]	Gain in flutter speed [%]
0.24	20	56	65	16.07
0	20	≥ 70	≥ 70	–
0.12	20	66	≥ 70	–
0.36	20	50	59	18
0.24	0	53	63	18.87
0.24	10	53	62	16.98
0.24	15	53	63	18.87
0.24	25	58	62	6.9
0.24	30	61	66	8.2

It can be seen that the sweep angle above 20 degrees makes flutter suppression more difficult, as the gain in the flutter speed increase drops significantly at these sweep angles. In addition, low flutter mass increases the open loop flutter speed as expected. Since the modeling and the control design was carried out between 40 and 70 m/s airspeed values, in some cases the flutter mode is not unstable for the given speed range. Similarly, in some case the flutter mode is stabilized up to 70 m/s airspeed, but this does not indicates the maximal achievable robust flutter speed.

7 Conclusion and Outlook

A summary of the interfaces, models and data formats that have been established in the MDO tasks for the demonstrator and scale-up workflow has been presented in this deliverable. The interfaces have been established over the course of the project enabling as automated of a dataflow as possible between the tools of different partners.

While the demonstrator workflow has been successfully run in the automated RCE environment, the scale-up workflow has at present been tested with manual handover of data through the established interfaces.

A selection of results from the demonstrator workflow is also presented, showing the benefits of such a multi-disciplinary automated workflow in performing top-level design studies catered towards a customized design objective.

8 Bibliography

- [1] Alberto Bemporad, N. Lawrence Ricker, and Manfred Morari. Model Predictive Control Toolbox User's Guide. 2022.
- [2] Raymond L. Bisplinghoff, Holt Ashley, and Robert L. Halfman. *Aeroelasticity*. Dover Publications, Inc., 1955.
- [3] Emilio M Botero, Andrew Wendorff, Timothy MacDonald, Anil Variyar, Julius M Vegh, Trent W Lukaczyk, Juan J Alonso, Tarik H Orta, and Carlos Ilario da Silva. SUAVE: An open-source environment for conceptual vehicle design and optimization. In *54th AIAA aerospace sciences meeting*, page 1275, 2016.
- [4] European Aviation Safety Agency. Certification Specifications for Large Aeroplanes (CS-25). 2007.
- [5] Tomasz Grabowski. PANUKL, 2022.
- [6] Vladimír Hanta and Aleš Procházka. Rational approximation of time delay. *Institute of Chemical Technology in Prague. Department of computing and control engineering. Technická*, 5(166):28, 2009.
- [7] J. Hofstee, T. Kier, C. Cerulli, and G. Looye. A Variable, Fully Flexible Dynamic Response Tool for Special Investigations (VarLoads). In *Proceedings of the International Forum on Aeroelasticity and Structural Dynamics*, 2003.
- [8] M. Karpel, B. Moulin, and P. C. Chen. Dynamic Response of Aeroservoelastic Systems to Gust Excitation. *Journal of Aircraft*, 42(5):1264–1272, 2005.
- [9] Thiemo Kier. An Integrated Flexible Aircraft Model for Optimal Control Surface Scheduling of Manoeuvre Load Alleviation and Wing Shape Control Functions. In *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Juni 2022.
- [10] Thiemo Kier and Gertjan Looye. Unifying Manoeuvre and Gust Loads Analysis Models. In *International Forum on Aeroelasticity and Structural Dynamics*, 2009.
- [11] Thomas Klimmek, Matthias Schulze, Mohammad Abu-Zurayk, Caslav Ilic, and Andrei Merle. cpacs-MONA – An independent and in high fidelity based MDO tasks integrated process for the structural and aeroelastic design for aircraft configurations. In *International Forum on Aeroelasticity and Structural Dynamics 2019, IFASD 2019*, Juni 2019.
- [12] Max Schwenzer, Muzaffer Ay, Thomas Bergs, and Dirk Abel. Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5-6):1327–1349, 2021.
- [13] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and design*. John Wiley and Sons, 2 edition, 2005.