FliPASED
WWW.FLIPASED.EU

Final Report

Authors: Fanglin Yu, Daniel Teubl, Thomas Seren, Christian Rößler (TUM), Charles Poussot-Vassal, Pierre Vuillemin, Nicolas Guérin (ONERA), Béla Takarics, Bálint Patartics, Tamás Luspay, Bálint Vanek, Bence Hadlaczky, Szabolcs Tóth, Milán Bárczi, Virág Bodor (SZTAKI), Matthias Wüstenhagen, Yasser M. Meddaikar, Thiemo Kier, Keith Soal (DLR)

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:	1.2 released on 13/12/2022
Start date of project: 01/09/2019	**Duration:** 46 Months

Lead Beneficiary for this deliverable:	SZTAKI
Last modified: 21/11/2023	**Status:** Delivered
Due date: 08/05/2023	

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

“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.”

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

Table of contents

1 Summary for publication

1.1 Summary of the context and overall objectives of the project

The FliPASED project is about developing multidisciplinary design capabilities for Europe that will increase competitiveness with emerging markets, particularly in terms of aircraft development costs. A closer coupling of wing aeroelasticity and flight control systems in the design process opens new opportunities to explore previously unviable designs. Common methods and tools across the disciplines also provide a way to rapidly adapt existing designs into derivative aircraft, at a reduced technological risk (e.g. using control to solve a flutter problem discovered during development).

The project aims to develop an advanced design toolchain and innovative methods for building a prototype aircraft that meets future aerial vehicle requirements. It combines simulation-based virtual design with extensive testing data from the EU FLEXOP project's flexible wing demonstrator aircraft. The primary research objectives are:

Improve the efficiency of separate wing design, flight controls, and avionics development toolchains by enhancing multidisciplinary collaboration for overall aircraft design. This approach optimizes fuel consumption through advanced flight control augmentation and uniform treatment of flight control surfaces. Develop accurate methods and tools for modeling flexible modes and synthesizing flexible aircraft control while ensuring reliable avionics system implementation, including fault detection and reconfiguration. This can lead to a 15% reduction in gust response amplitude and standardized methods to streamline mathematical models, reducing engineering effort by 50%. Validate the developed tools and methods using the experimental platform from the prior H2020 project (FLEXOP). This platform enables interdisciplinary development, testing gust response prediction, active wing morphing for fuel efficiency, and flight envelope assessment in failure scenarios. Significant cost savings are expected by manufacturing advanced flexible wings and reusing main components from FLEXOP, expanding the design space for active flexible wing capabilities and generating valuable data for research and industry through open data sharing.

1.2 Work performed from the beginning of the project to the end of the period (M1-M54) covered by the report and main results achieved so far

Work has been performed in four technical and one management work packages, while WP1, WP2 and WP3 started early in the project in the first 24 months only minor preparatory work was done on the scale-up work package (WP4). During the second half of the project scale-up and the corresponding RCE toolchain development was more active, but the incident with the demonstrator and numerous re-planning activities due to covid related travel restrictions made the work distrubution and progress far from ideal. This also led to re-focus the project and abandon the -3 wing manufacturing and flight testing. The main work items and achievements were the following:

- Setup of requirements incl. open data process,
- Definition of collaborative work process including interfaces between disciplines & selection of collaborative work tools all within the RCE environment and interfaced via standard CPACS data exchange
- Reference model definition

- Enhancement and maturation of (single discipline) tools towards robustness
- Demonstrator overall and component level improvements
- Setup of integrated tool framework (inc. design of control functions)
- Model refinement using GVT data & flight tests
- Setup of a collaborative (remote) workflow for many sub-problems
- Rebuild the demonstrator
- Flight test the demonstrators (T-FLEX and P-FLEX) and conduct flutter related experiments
- Publish datasets openly and disseminate the project results

WP1 The demonstrator MDO workflow with interfaces and requirements were set-up. The wing and demonstrator actuation and sensing concept was reviewed to account for the increased need of sensing and larger number of actuators coupled with the main objectives of demonstration. Also, the requirements were reviewed to show clear benefits for a/c MDO design, where different advanced functions must work together in the design phase and their improvement potential has to be quantified.

- Workflow is defined and the workshare between partners is established both for demonstrator and scale-up
- A good compromise between open access and high fidelity aircraft model have been found, the required components and suitable features have been selected
- Components via standard interfaces have been connected and their interaction has been tested
- Matlab based tools have been improved and tailored for better reuse in connection with flight test data analysis, OpenMCT based real-time flight test data analysis tools have been adopted, Nastran deck based data sharing for model update, common HIL platform for control testing and pilot training
- Data sharing during GVT, RCE runs and flight testing – including model updating procedures in Matlab and NASTRAN established. Flight test data published on https://science-data.hu/dataverse/flipased
- Conceptual design for sensing and actuation of -1 and -3 demonstrator fixed: Lots of benefits from acc + gyro and control surface deflections Improvements of existing functionalities on the demonstrator are proposed and implemented
- Required ground and flight test schedule updated
- Avionics redesign and improved ground control station interface
- Several iterations of more seamless post-processing of flight test data for improved flight path and flexible mode reconstruction
- MDO toolchain objective definition and definition of demonstrator / -3 vs. Scale-up aircraft cost function
- MDO workflow definition
- Agreement on tools and adoption of them within the collaborative work process (RCE, CPACS, NASTRAN, Matlab, Python, Tixi, Tiegl, Nextcloud, Overleaf, etc.) Facilitating analytical modelling and physical test based updates and iterations Comparison of model based and data-driven methods to capture aeroservoelastic behavior

WP2 The demonstrator MDO workflow with interfaces and requirements were adopted and the individual tools were given a common RCE/CPACS interface. Building and intergrating methods/tools was successfully consolidated. In addition to the individual building blocks within the RCE toolchain new modules were also developed without integrating them into the toolchain.

- Model step
- LTI (reduced) model construction of the demonstrator, including sensors and actuators
- LPV (reduced) model construction, with additional uncertainty modelling on flexible parameters (modal freqiency and damping)
- Control design architecture setup for various functions
- Flight controller to achieve baseline handling qualities and perform repeatable flight test experiments
- Load controllers for maneuver and gust load alleviation
- Flutter controller using two different approaches (input-output blending by DLR and structured DK-synthesis by SZTAKI)
- Analysis of closed loop performance including robust flutter speed and margins
- Fault detection of actuator and sensor faults
- Comaprison of analytic (EKF) and data driven wingshape estimation (KalmanNET)
- Wing drag and lift calculation and a comparison of various tools
- Wingshape control for (induced)drag minimization

WP3 is related to connecting theory to practice. There were some major deviations in the original plan due to Corona restrictions on shop floors and the airport, as well as landing gear problems had to be solved before the flight tests.

- Demosnstrator prepared for flight testing with new landing gear, telemetry and sensor setup
- Building the advanced -3 wing: backward planning and Plan B established
- Ground testing the demonstrator: GVT, Static test, HIL simulations, Flight Test Crew training
- Flight Test Specification and System Identification: Flight phase #1, #2, #3 planed and executed
- Gathering 600 minutes of flight test data at various conditions with extensive measurement suite.
- Developing the actuator for flutter control.
- Validating the flutter prediction tools in real life
- Displaying successful functioning of two active flutter controllers in real-life conditions
- Sensor concept refined
- Sensing concept for new wings
- V-tail IMUs
- Aeroprobe and IMU repositioning

- Thrust measurement
- Demonstrator wing design: -3 planing and back up Plan
- Demonstrator component upgrades:
- On board computer 2
- Open MCT (Flutterometer)
- Antenna Upgrade
- Landing gear improvements and testing
- Power wiring
- Flight test specification and system identification
- Flight testing

Within WP4 the scale-up task is slightly different from traditional MDO workflows: FliPASED has a strong Focus on the control part in the MDO exercise and not the aero-structural optimization. A comprehensive loads process with an actively controlled flexible aircraft is in the loop covering a comprehensive set of load cases, unlike many MDO chains with CFD in the loop. The process is implemented as a distributed collaborative process. Due to low fidelity aerodynamics no HPC resources are needed.

Due to this some limitations are expected: Only low fidelity / potential flow aerodynamics are used, therefore no transonic effects are modelled. Wing shape control functions only account for induced drag, i.e. no shock control by camber changes to reduce wave drag. Flutter mechanism and Active Flutter Control without transonic dip. However controller synthesis remains the same.

- Reference model (DLR D150) selected
- Workflow setup for the scale-up task including the free parameters within the investigation
- Mission criterion setup
- Objective function definition
- Development of the individual control building blocks
- Connect the open-loop and closed-loop loads models and the corresponding sizing
- Implement wingshape control and analyze the tradeoff between flap size and achieved benefits
- Performance evaluation

WP5 The demonstrator MDO workflow with interfaces and requirements were set-up. Setup of the project management and collaborative environment for the project is complete. Publicatios and exploitation is tracked and managed within the consortium.

The consortium established collaborative tools for project management (Nextcloud + iterative GANTT charts), software development (Git), document editing (Overleaf). Moreover, the collaborative work process also involves common hardware development tools - a common hardware-in-the-loop platform which is shared across DLR, TUM and SZTAKI. The partner contributions within the common MDO toolchain are all implemented and tested using the RCE environment.

Large number of publications and other dissemination activities were pursued, including engaging not only with the scientific community but also with the general public.

The consortium also succesfully managed the impact of Covid, chip shortage and the replanning associated with the loss of the first demonstrator - leading to a 6 months ammendment.

- Management of the partners contributions
- Reporting towards the EC
- Organizing meetings online (during Covid) and in-person
- Disseminate results in workshops and conferences
- Manage an amendment due to the loss of the demonstrator
- Setting up collaboration tools between partners

1.3 Progress beyond the state of the art and expected potential impact (including the socio-economic impact and the wider societal implications of the project so far)

The primary impacts of the project are related to the succesful active flutter control tests on a conventional configuration UAS, the release of a validated datasets to the aeroelasticity community within the Open Research Data initiative and the improved design environment comprising enhanced toolsets optimized for collaborative interactions within and across organizations, together with the best practices and standards for collaborative designenvironments as well as the design process itself. Since the developed tools are validated using a flight demonstrator, the reached Technology Readiness Level of this activity is TRL5. Through dissemination of validated data publicly, the project will provide a rare opportunity to generate great impact on the flexible aircraft research community as flight test data are very sparsely available due to confidentiality reasons. Subsequent impact will be achieved through the application of the enhanced design process on new aircraft development activities, leading to significant improvements in development and certification costs, providing continuous progress towards improved structural and aerodynamic efficiency - a trend very visible with the recent introduction of high aspect ratio demonstrators by both Airbus and Boeing. This technological progress has an impact on airline operating costs through fuel savings, leading also to the environmental goals set out in the ACARE goals and Flightpath 2050.

In particular the project went beyond the state-of-art in several topics. The overall RCE environment aimed for MDO tasks, developed jointly between the partners and taking flutter control into consideration is a true novelty - the parametric design study with the ability to show the efficiency of flutter control at different parts of the design envelope is very novel.

Several of the building blocks within the tools are also significantly beyond state-of-art. Most noteable the aeroleastic modelling toolchain including uncertainties in structural frequencies and damping, the corresponding model order reduction methods and the subsequent flutter control design framework, including the ability to update the design models from GVT and flight test related updates.

The aircraft avionics and sensing system also includes several novelties, with the increased reliance on angular rate measurements besides accelerometers. It is clearly visible from the flight test data how much additional information can be subtracted from the wing torsion related modes with torsional gyroscopes.

The automated MLA and GLA design framework, where control design objectives are also automatically tuned is also required for automatic MDO processes where control is part of the optimization, and it includes several novelties.

The partners also developed, ground tested and flown a miniature operational modal analysis system, what could be a game-changer for not only the demonstrator but also applicable for eVTOLs, general aviation aircraft as well as larger drones.

The achieved results related to active flutter control needs no further explanation. Several projects, including ones led by Airbus and Dassault Aviation are looking at very similar problems and the project directly supplies information to them since both Airbus and Dassault were on the advisory group of the project.

Novel custom actuator system were developed within the project to actuate the flutter flaps, these are some of the highest frequency aerodynamic actuators fitted to a similar sized drone and required significant engineering effort beyond the state-of-art to manufacture and integrate them to the demonstrator.

The project also spent significant effort on Big Data based methods to provide a Machine Learning based solution to wingshape estimation (an essential component to wingshape control based drag reduction) using the KalmanNet architecture, and compare its results to traditional Eyxtended Kalman Filtering methods.

Fault detection and isolation methods for flexible aircraft are also very new, there are only one additional article we were able to find where FDI methods and flexible dynamics are considered in the same framework. The FDI experts from Airbus have invited the team members to the IFAC Safeprocess conference to present the results about developments within FliPASED.

The project was the first in Germany and one of the first in Europe under the new EASA regulations to have flight authorization for a more than 25 kg fixed wing drone flying in commercial airspace. Both LBA and DLR Cochstedt Airport were very constructive and they also benefitted greatly from the approval process.

Many MDO projects have considered aerodynamics and structures coupled optimization, but only a handful of them considers maneuver and gust loads, hence the framework set up within the project for loads closed sizing loop for a commercial aircraft is also very novel.

System identification methods tailored for flexible aircraft model update are used by flight test teams within the aerospace industry, but the novel features developed within the project allow reconstructing all the measured responses from the flight test with high accuracy, moreover the updating method can be employed in a single step with different databases provided from flight tests with nearly identical initial conditions, which would lead to a more physically realistic correction of the system matrices.

TUM gained significant expertise in conducting flight tests with large UAVs, and also in designing aeroservoelastic demonstrator wings especially for larger aeroelastic response, including flap numbers and tuning masses for joint structural/aerodynamical optimization. With this expertise TUM is able to maintain its leadership in aircraft design and its knowledge on lightweight structures would secure them further collaborations in aeronautics. The scientific results which will be published as a part of PhD Theses will have significant impact on the scientific community. Also TUM developed an extensive testing environment, including pilot training, visualization, HILS testing, airport operational procedures, which makes them outstanding in a worldwide UAS research and testing field. The succesful flight permit process with LBA and EASA also gives them a significant advantage against the other universities around europe in the +25 kg fixed wing drone testing field.

SZTAKI gained significant expertise in active flutter control design, implementation, testing and deployment. Related to this SZTAKI also gained insight on the performance trade-offs with respect to

sensor selection, avionics delay and update rate, as well as actuator limits on the achievable control performance. The related control oriented model order reduction of large scale LPV systems is yet another novelty. With this expertise SZTAKI will be able to maintain its leadership in LPV systems theory and would secure them further collaborations in various fields. Within the project a modelling and control design framework was set-up and matured to be applicable for general flexible body dynamics problems. A structured modelling and structured control design framework was developed within the project, which is appealing to industrial practice due to the resulting low-order controllers, while able to handle plant uncertainty not just only nominal behaviour, what is also outstanding in the control scientific community. Control system implementation aspects have been significantly progressed by SZTAKI, developing its hardware/software framework for automatic code generation, what will be utilized in many research projects beyond flight control even in autonomous ground and surface vehicles. The scientific results which have been published as a part of PhD Theses will have significant impact on the scientific community.

ONERA gained significant expertise in conducting GVTs and relating its results to flutter prediction, while all the relevant data are shared open-source providing an opportunity for further analysis and collaboration with both academic and industrial partners. Related to this the GVT experiments were also augmented with built in sensors and the experiments were conducted with flutter control system active and inactive. Hence, ONERA gained significant insight on conducting tests with flutter control systemes operational during the tests. ONERA also gained significant expertise in automated MLA synthesis process with the corresponding automated aeroservoelastic model order reduction methods and the relationship of this process to the GLA and baseline controller synthesis.

DLR gained significant expertise in active flutter control design, implementation, testing and deployment. Related to this DLR also gained insight on the performance trade-offs with respect to sensor selection and control law update rate, as well as actuator limits on the achievable control performance. DLR also gained significant insight in MDO problem setups both on the demonstrator and on the commercial aircraft scale, where flexible structures with pre-defined control layout, aerodynamic models and control laws have to be included in the design. This expertise will help them to become more experienced in aeroservoelastic conceptual design rather than the traditional aeroelastics discipline, which in turn helps DLR to produce more relevant scientific results for both structures and flight controls communities. DLR also matured its tools and methods to work in a collaborative fashion with other academic and industrial stakeholders as a partner who aggregates all the technical results from the partners and builds a complete aircraft model, taking into account the various interfaces, modelling fidelity and design language of various groups. The project also provided an opportunity to develop, flight test and further refine in-flight operational modal analysis methods and tools what could be key enabler in the future flexible aircraft development. During the GVT tests DLR scientist also worked on structures and active control coupling mechanisms similar to industrial test campaigns. The project results provide an excellent opportunity to share these findings with a publishable dataset, without company confidentiality.

2 Explanation of the work carried out by the beneficiaries and Overview of the progress (Technical Report 1)

2.1 Explanation of the work carried out per WP - Work Package 1

WP1 (Recommendations capture and attainment) focuses on derivation of key requirements for the aircraft and demonstrator under investigation as well as the interconnection of design tool and the relevant data acquisition and analytics methods and process.

2.1.1 Objectives and activities

The main objective of this work package is to address the complete integrated avionic process including aircraft shape, sensors and actuator locations and detailed control design. The purpose is to set up an integrated collaborative framework and tool-chain for the design of a new passively and actively controlled flexible wing-based aircraft, in a safer and more reliable context. The purpose of the activity is to end-up with an enhanced and fastened maturation process tool to quickly reach high maturity levels through digital-based methods and tools.

- Detailed design of control functions
- Enhancement and maturation of (single discipline) tools
- Setup of integrated tool framework
- Establish integrated, collaborative design tool chain
- Re-Design of FLEXOP -1 Wing established (Validation)
- Design of new advanced active controlled wing
- Establish redundancy based methods for enhanced safety and reliability
- Connect the theoretical toolchain and the flight test goals

2.1.2 Starting point and approach

The two main starting points for this WP are the existing demonstrator design (inherited from FLEXOP), what has to be accommodated into the improved design toolchain, and the existing tools, standards and available scale-up aircraft models of the partners.

The approach consists of assembling the main specifications and criteria what must be demonstrated during the flight tests of the demonstrator and from these specifications the required hardware and software modifications and improved data processing workflow have to be developed in an incremental fashion.

On the other side, the limitations of the current demonstrator has to be established (like low Mach number, lack of wind tunnel and complex CFD testing). Based on these limitations the scale-up study also has to limit its scope, to be aligned with the tools and methods developed by partners.

Besides the existing tools the partners have to agree also on the common standards and workflow, what is highly facilitated by DLR's experience with MDO tools and the related CPACS data interchange language.

2.1.3 Efforts and achieved results

The main effort in WP1 was to set-up the tasks and responsibilities for the MDO workflow. This involved breaking down the conceptual design of the demonstrator into design steps and assign a responsible partner within the consortium to each one of them. This involved creating clear performance objectives and agreement on the data sharing format (CPACS). The overall workflow (Figure 1) will be executed on a distributed cluster of workstations, scheduled by RCE environment (Figure 2).

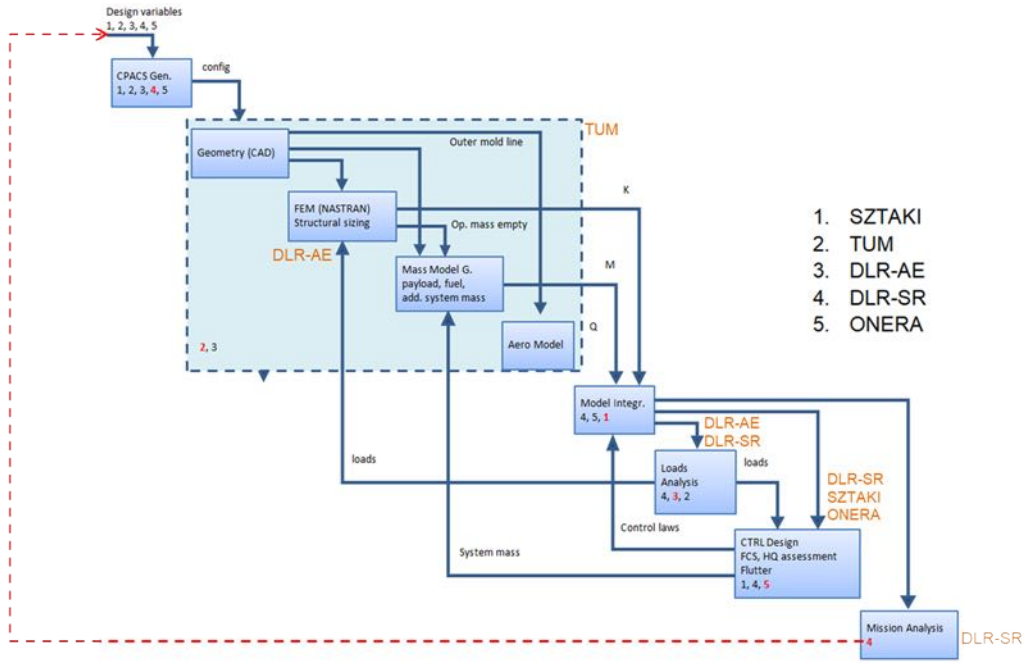
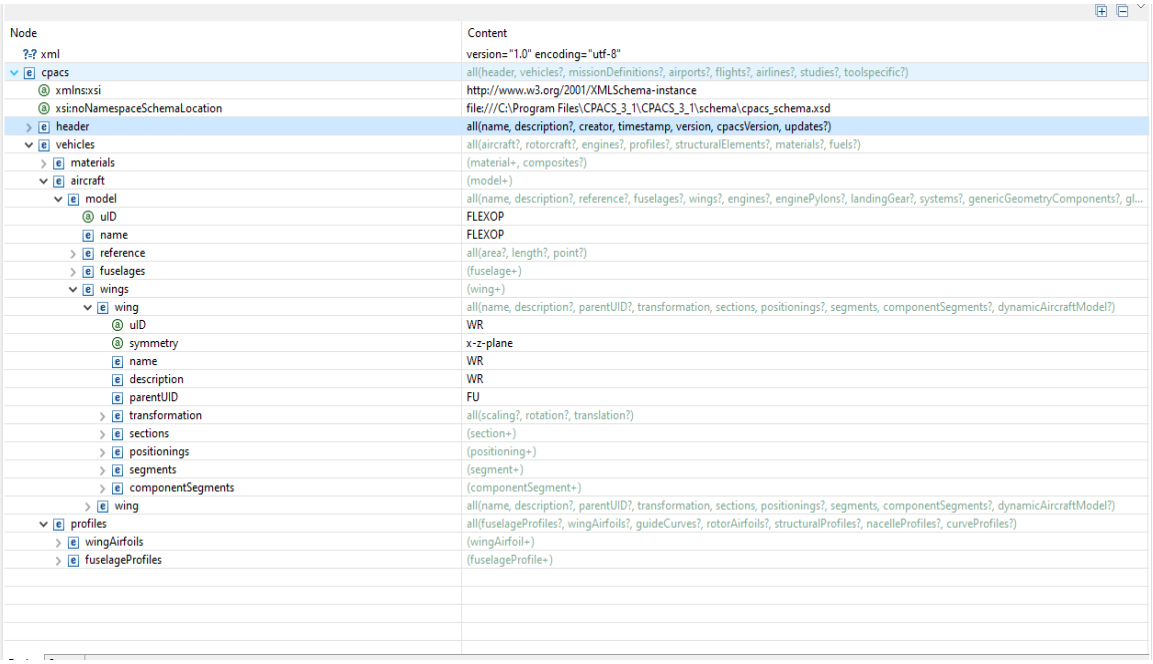


Figure 1: Demonstrator workflow

Agreement on tools and adoption of them within the collaborative work process (RCE, CPACS, NASTRAN, Matlab, Python, Tixi, Tiegl, Nextcloud, Overleaf, etc.) was natural, but required longer time since IT infrastructure was inaccessible for long time due to the pandemic.

Roles within the workflow are the following:

1. CPACS basic description generation is performed by DLR using Matlab based tools, with visualization aids from Tixi/Tiegl (Figure 3)
2. The aero-structural block is handled by TUM with the help of DLR
   (a) Aircraft external and internal (structural) geometry is generated with the Phyton based CAD tools made by TUM, leading to parametric description of the wing – with 3 different versions of distinct flap numbers: 4, 8, 16 per wing.
   (b) The FEM model is generated automatically in NASTRAN by the tools of DLR-AE
   (c) The aero model is based on standard VLM/DLM methods, but the teams of DLR, TUM and SZTAKI are working on increasing the fidelity of induced drag prediction to include its effect in drag reduction control law development
3. Model integration is done in Matlab by DLR-SR based on the standardized components coming from the aero-structural blocks



Node	Content
?-? xml	version="1.0" encoding="utf-8"
cpacs	all(header, vehicles?, missionDefinitions?, airports?, flights?, airlines?, studies?, toolspecific?)
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation	file:///C:\Program Files\CPACS_3_1\CPACS_3_1\schema\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?, gl...
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+)
wing	all(name, description?, parentUID?, transformation, sections, positionings?, segments, componentSegments?, dynamicAircraftModel?)
profiles	all(fuselageProfiles?, wingAirfoils?, guideCurves?, rotorAirfoils?, structuralProfiles?, nacelleProfiles?, curveProfiles?)
wingAirfoils	(wingAirfoil+)
fuselageProfiles	(fuselageProfile+)

Figure 2: Visual representation of the XML based CPACS file describing the FLiPASED demonstrator

4. Loads analysis is performed by the tools developed at DLR-AE, which is a mere check in the demonstrator workflow but will be fed back to the FEM/Structural sizing in the scale-up.
5. The various control design related components are assembled in the workflow int one functional block, overseen by ONERA. This block includes several functional sub-components:
   (a) Model order reduction (ONERA & SZTAKI)
   (b) Baseline controller synthesis (SZTAKI)
   (c) Maneuver load alleviation (ONERA)
   (d) Gust load alleviation (DLR-SR)
   (e) Flutter control (SZTAKi & DLR)
   (f) Drag reduction control (SZTAKI)
   (g) Stability and HQ assessment (SZTAKI)
6. The overall aeros-servo-elastic system is analysed and performance is calculated in the Mission Analysis block, led by DLR-SR.

An objective function was also carefully selected for the demonstrator workflow to provide focus for the optimization (Figure 4) – we aim to maximize the weighted sum of open-loop vs. closed-loop difference in drag, flutter speed, maneuver load, and gust load. This would provide the benefit of being able to demonstrate the highest contrast between open-loop performance vs. closed-loop performance. A wing/demonstrator designed in this way would be flown with control laws switched off and the performance recorded, then the control laws would be turned on one by one and the increase in performance would be assessed.

Within the reporting period a number of changes have been proposed to improve the conceptual design of sensor layout and actuation system of the wing (3), and the improvements made and proposed on the fuselage based on operational experience.

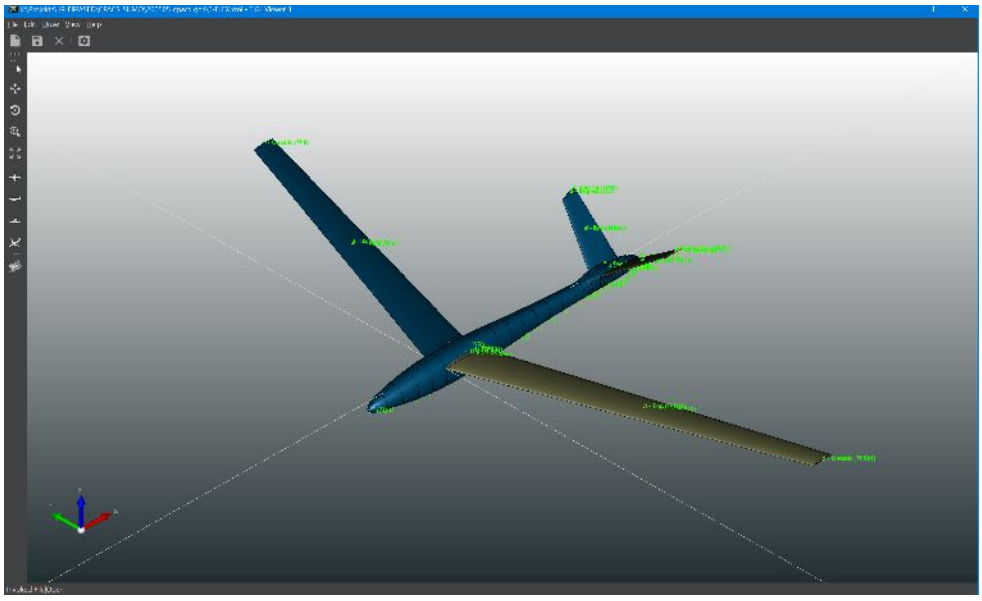


Figure 3: Screenshot of FLiPASED CPACS File with TiGL Vizualization library

Starting design
-1 wing from FLEXOP
Fuselage
V-tail

Design freedom
Wing planform (sweep, taper ratio, span)
Spar positions
Skin stiffness
Number of flaps (with parallel models)
Flap positions
Wing pre-twist
~~Number~~ and position of ribs
Material choice (?)

Objective function
max(a*(Drag_OL-Drag_CL) + b*(Flutter_OL-Flutter_CL) + c*(ML_OL-ML_CL) + d*(GL_OL-GL_CL) + 0.0*Weight)

TLR
Flutter_OL < 50m/s; x Hz
~~Flutter mechanism~~
Weight_ac < 65.0kg

Figure 4: Optimization cost function and design parameters

During integration and operation of the aircraft the operation team found a couple of design related problems which either made the operation unsafe like the landing gear, or give too harsh boundary for critical function implementation like the lack of digital remote control interface on the RX-MUX units. Along with that, some additional changes were already made to improve the existing functionality like secondary on board computer, and further changes are proposed to have even increased functionality like electrical power measurement or High bandwidth telemetry system. The deliverable D1.1 introduce these changes in more depth.

The wing (3) sensor layout and actuation system has been also revides and a number of improvments have been proposed and implemented. Along with the inertial measurement units used on previous wings other sensor layout concepts are proposed. On flight control and actuator system side, a CAN bus based actuator system is proposed. Along with that a detailed comparison is given between the proposed design and the system used on previous wings during the legacy FLEXOP project.

The experiences and detailed study on the servo health monitoring system currently used in the -0, -1 and -2 wings have been also revised and improvements have been proposed to increase servo deflection measurements for better system identification.

The main contributions of the team are:

- Collecting the main changes proposed in the fusealge, compared to original design documents (Figure 204).

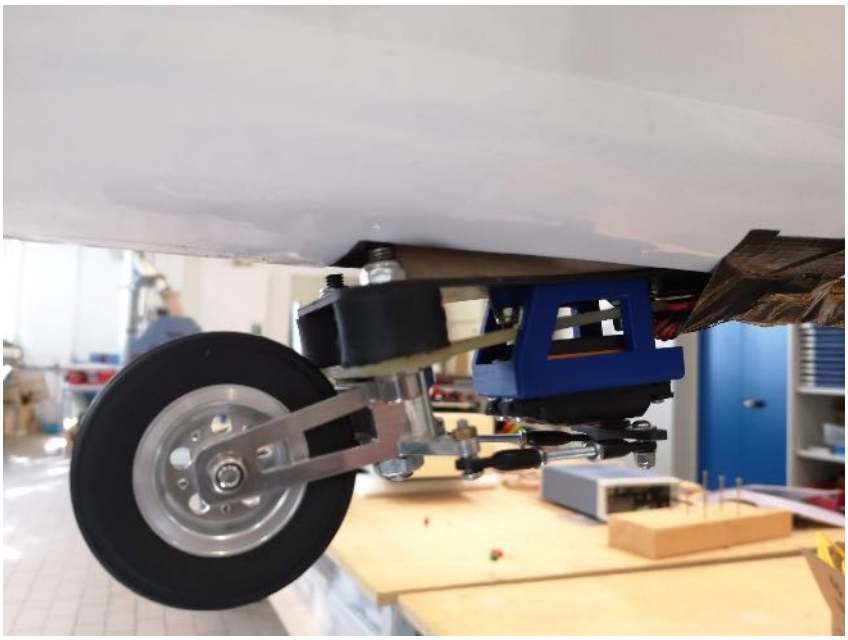


Figure 5: Modified steerable landing gear

- Providing an updated sensor layout concepts for wing (3) (Figure 6).

- Providing an actuator system concept for wing (3).
- Showing a detailed analysis of the previously used servo health monitoring tools system (Figure 7).

On the overall MDO design loop front to set up a collaborative design toolchain for an advanced, actively flight condition optimized wing design, requirements for the MDO toolchain need to be captured first. Deliverable D1.2 captured the outcomes of activities conducted for the requirement capture and serves as the top-level guideline for the subsequent MDO implementation.

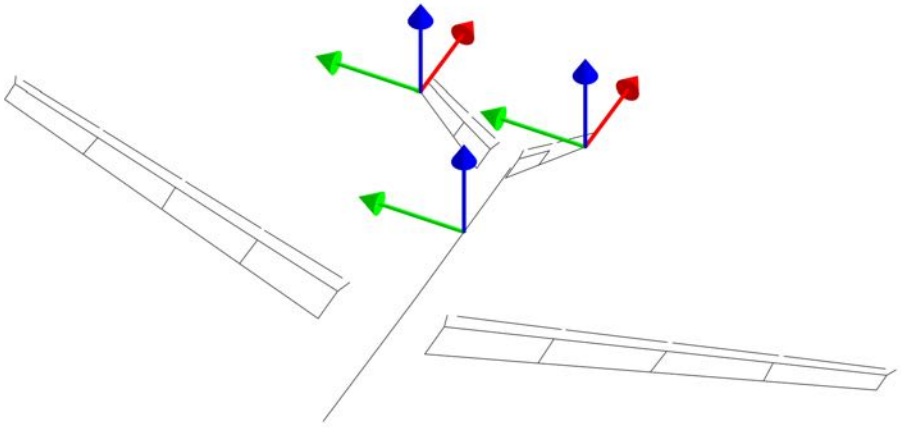


Figure 6: Improved onboard IMU locations for tail flexible motion detection

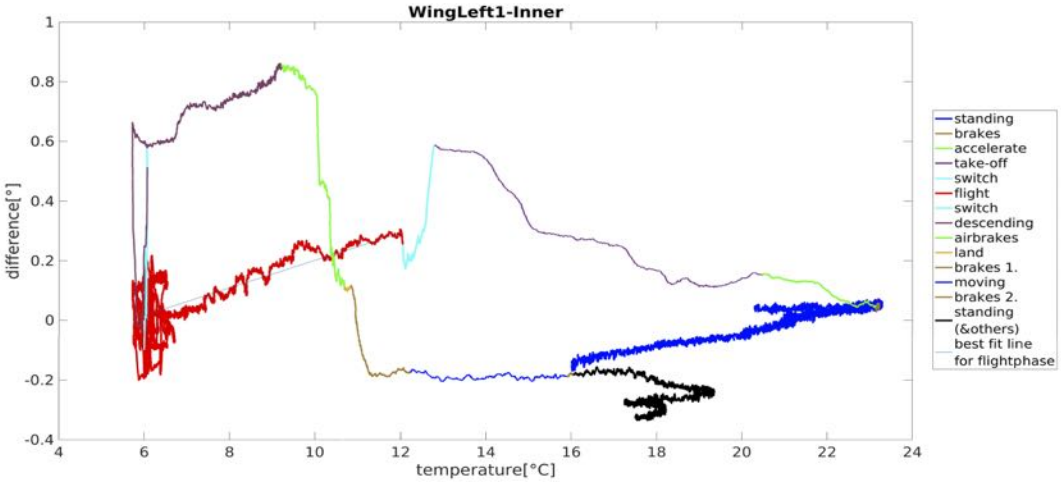


Figure 7: Temperature and flight state related calibration of servo deflection feedback

The tasks conducted within the period related to setting up the overall collaborative MDO toolchain have been the following:

The objectives of the MDO toolchain (Figure 8) and derived requirements were discussed and agreed among the partners. Two sorts of requirements are specified because of the different objectives for demonstrator wing design and commercial transport aircraft wing design.

The context of the consortium's activities related to the industry standard MDO toolchains were studied. Based on prior project results and experience the MDO toolchain structure is captured by MDAx, which is developed by DLR to support the ideation phase of MDO. The functions of individual blocks are specified and their interconnection has been iterated among the partners. An introduction of the integration framework RCE is given here (Figure 9).

The definition of interfaces of connected blocks in MDO toolchain required significant effort, due to the multidisciplinary nature of the project and due to the need that each block has to be 'human intervention free', to avoid lengthy hand tuning of parameters by experts within the MDO iteration loops (Figure 10).

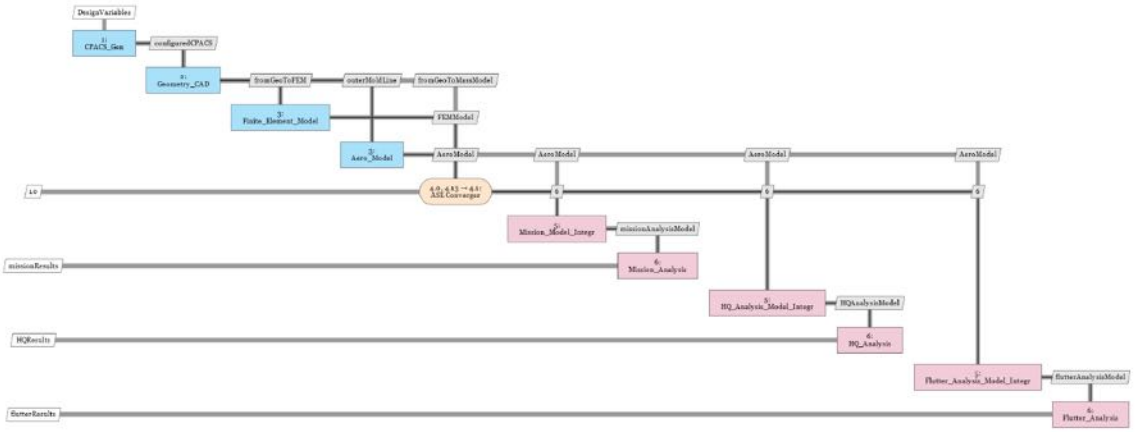


Figure 8: MDO worklflow in XDSM format with collapsed 'ASE converger'

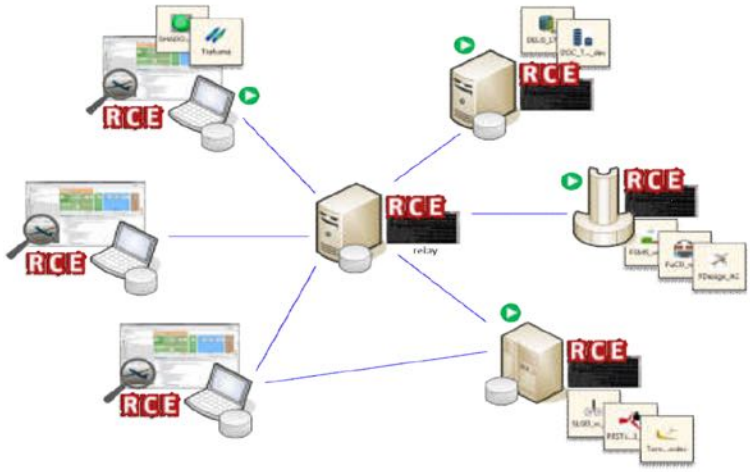


Figure 9: Distributed RCE workflow

An introduction to CPACS, which is agreed by the consortium to serve as the standard interface medium, is also given here.

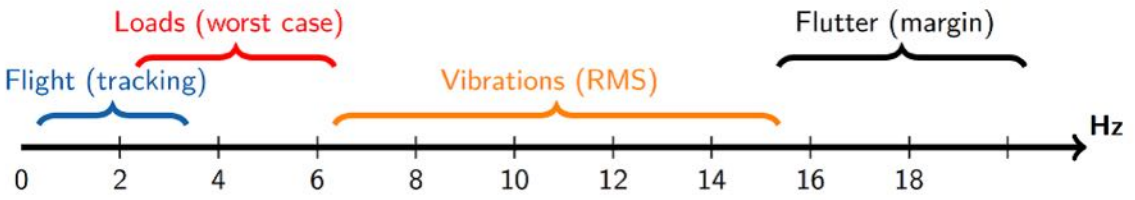


Figure 10: Frequency grid of the physical phenomena occurring over an aircraft. Ranges and values are different from an aircraft to an other

Demonstrator Airframe

The two goals of the legacy FLEXOP project, namely aeroelastic tailoring and active flutter suppression, presented very different requirements for flight tests [114]. Testing of the aeroelastic tailoring would require high load factors on the wings and were not expected to pose any operational challenges. On the other hand, flutter testing demands high airspeeds and, therefore, big areas for manoeuvring (acceleration, deceleration, high-speed turns). Additional requirements were placed by the scale-down task, imposed by industrial partners (geometry similar to a new generation commercial airliner), sensors required for the measurements (minimum 2kg of payload capacity) and limitations due to logistics (maximum part size of the unrigged aircraft should not exceed 4m).

Based on these requirements a flight test mission was designed and, including the UAV design experience of TUM, a preliminary design of the demonstrator was done. This resulted in a 65kg take-off weight (TOW) demonstrator with a swept, 7m span wing and a V-Tail. The demonstrator received three pairs of wings: the rigid wing for setting the baseline (designated as -0), the wing with active flutter control (-1) and the aeroelastically tailored wing (-2). Risk alleviation by system redundancy was incorporated for aircraft controls. The concept required symmetrical control of the aircraft even if one of the batteries powering the aircraft control surfaces would lose voltage. This requirement resulted in 8 wing flaps (4 per wing) and 4 ruddervators (2 per V-Tail). Additionally as a last measure to protect the infrastructure, a parachute within the tail part of the fuselage was integrated.

Main characteristics of the demonstrator can be found in figure 19. The demonstrator is mostly flown

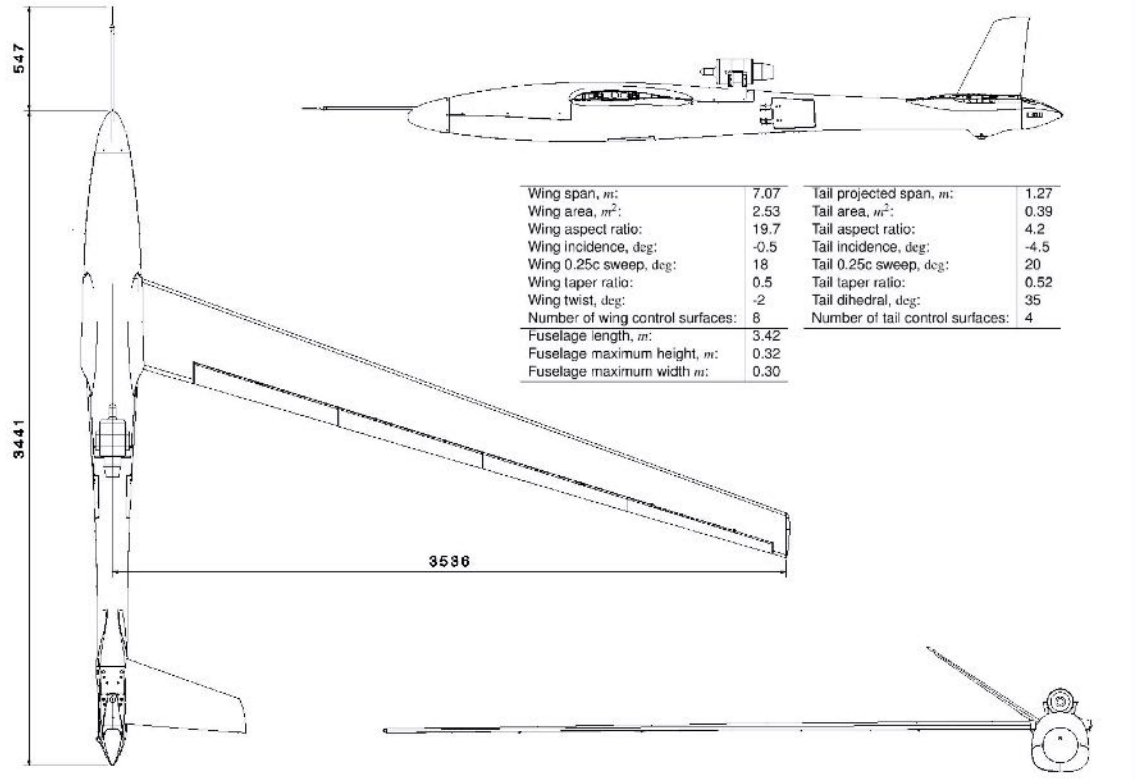


Figure 11: FLEXOP Subscale flight demonstrator. Note that the left wing and V-tail are excluded.

manually by pilot via external vision. Stability augmentation flight mode via angular rate feedback is

also available, where in manual mode the surface deflections are directly linked to the joystick positions on the transmitter and the measured angular rates.

The aircraft has two control links. Control via two different transmitter brands was desired to decrease the risk of both transmitters failing together due to a common mode failure (either connectivity, electrical or mechanical). The main one is a Jeti DS-24 system which has an additional back-up receiver that is integrated further away from all the other radio links. The secondary RC system, controlled by the backup pilot, is a Graupner mc-28 system. Graupner has only one receiver with four antennas that are pointed in different directions. In comparison, Jeti receivers have only two antennas.

Propulsion System The main requirements while designing the propulsion system were high acceleration, low vibration and precise speed tracking. Taking these requirements into account, a jet engine paired with a fast-response airbrake system was selected. The jet engine is a BF B300F turbine with 300N maximum thrust capability. The engine was mounted on a pylon above the fuselage with the fuel tank located directly below it. This was designed with the intent to keep the same centre of gravity throughout the flight.

Other systems A 5-hole air-data probe provides the measurements of aerodynamic angles and airspeeds, as well as static and total pressures. The measurements are captured within the Micro Air-Data Computer manufactured by Aeroprobe. The probe is mounted on a boom 55cm in front of the demonstrator nose. The boom length was determined using the airflow data received from the Computational Fluid Dynamics (CFD) simulations. During the simulations, airspeed and flow angle values were compared at different distances away from the nose. The distance which resulted in local flow values within 1 percent of the free-stream values was chosen.

A secondary airspeed reading is measured by a low-cost air-data probe mounted on the right V-Tail of the aircraft. To make sure that the readings on the secondary air-data probe are satisfactory for backup operation, the calibration of the probe was checked in the wind tunnel. Furthermore, the airspeed measurements in between the two probes were compared during the first flight of the demonstrator. Good correlation of both measurements gave confidence that even in the case when the main airspeed sensor is lost, a reliable backup would be available.

The position and attitude of the aircraft is measured by a high-precision Inertial Measurement Unit (IMU) MTi 710 manufactured by xSens. Additionally, multiple IMU units were installed in the wings for capturing structural acceleration data during flutter testing. Additionally, fiber Bragg grating (FBG) system was installed in the wings for accurate deflection measurements.

A parachute system, comprising of a drag chute and the main chute, is installed in the aircraft (manufactured by skygraphics AG). In case the chute release is triggered, the magnet, holding the tail cone, is released. The tail cone is then pushed away by the incoming airflow. It has the drag chute attached to it, which, consequently pulls the main parachute out.

Two small cameras (Mobius 1080p HD Action Camera) are integrated within the tail cone. The cameras were placed in a way to overview both wings in-flight and provide visual feedback after test runs. They were not accessible online and would only be used for offline evaluation.

Interior layout of the systems is displayed in figure 20.

Demonstrator avionics The avionics system was built from the ground up to serve the custom needs of the demonstrator. No commercial off the shelf system would provide the required number of input interfaces (custom sensors, RC, telemetry, etc.), as well as the output interfaces with 19 PWM servos, and custom UART based propulsion unit interface. The research task of integrating custom, highly sophisticated, modern control and estimation methods, instead of the standard PID gains also facilitated the need of a custom avionics solution.

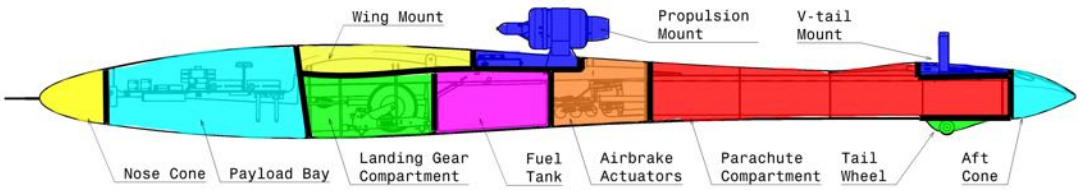


Figure 12: Interior layout inside the demonstrator.

Autopilot

The flight tests require to test the autopilot features one-by-one, i.e. switching on one loop after another and observe their behavior at the flight test campaign. In order to handle different mission objectives, the pilot should have to command GCS to set the FCC to switch to the desired operation mode. This command is basically sent out from the GCS, and then the pilot decides when to perform it. Therefore, two autopilot levels are distinguished. The Autopilot 1 (AP1) mode is just an augmented mode at most of the time, and in this mode, the state machine inside the autopilot receives the commands from the ground. The Autopilot 2 mode (AP2) performs the preset action, when the pilot decides to switch to it on his 3-level-switch on the JETI transmitter.

Mission Planner interface

The GCS employs a customized Mission Planner which has selectors for the operation modes, what are sent to the FCC via the telemetry antennas. The options displayed by the Mission Planner depend on the capabilities of the software running on-board[1]. Figure 21 illustrates the custom user interface in Mission Planner.

As you can see, there are options to turn on or turn of logging on the FCC and multiple options to alter the functions of the autopilot.

Implementation

The state machine is implemented inside the autopilot's Matlab Simulink model, as a Stateflow Chart, so the code generation from the implemented autopilot logic can be done with the well-known Matlab Simulink Coder. The mask of this state machine in Simulink can be seen on figure 22. The augmented mode is active in all autopilot modes (both AP1 and AP2), and the actually flight tested components are switched on in AP2 mode. This can involve inner loop functionalities, and task specific modes. After a successful test flight with a certain software, the next software's autopilot mode AP1 can include the capabilities which are proven to work in AP2 mode. Then, the flights can be basically performed with switching to AP1 after takeoff, then using AP1 until landing, and switching to AP2 for performing a Mission Planner specified objective. The details of the performed maneuvers are described in the Flight Test Cards.

Another novelty in FLiPASED is that telemetry parameters are logged on the FCC. Only essential parameters were logged previously, what caused the lack of information, so we could not reconstruct the entire chain of events during a test. From the log we could not determine in which state the autopilot was (even though they were logged on ground), which button has been pressed on Mission Planner, and when. Therefore now the parameters which are set by the GCS are logged onboard.

[1] At a flight test, the flight test crew should be helped by the user interfaces as much as possible. The software on FCC is started with a custom USB pendrive solution, which also selects the autopilot version. This version is reflected at the GCS, therefore only appropriate command messages can be sent out. For example, if a software is dedicated to run engine identification, Mission Planner displays different options than an another software, which is used for injection of modal identification signals. On Figure 21 you can observe the configuration modes and buttons for baseline mode functionality.

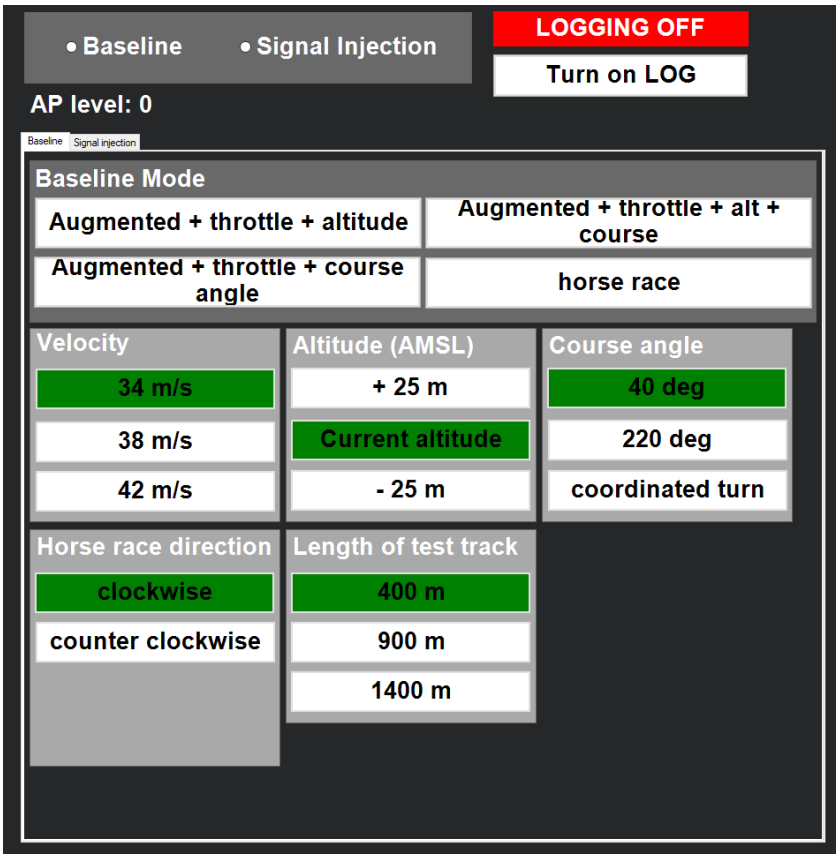


Figure 13: Mission Planner user interface

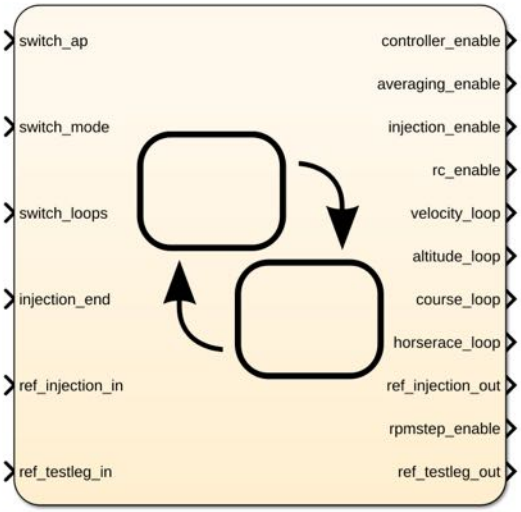


Figure 14: Matlab Simulink StateFlow Chart inside the autopilot

ECU communication, fuel flow display in EDL

The signal of the fuel flow sensor provides a value proportional to the rate of consumed fuel in each measurement step. Thus, scaling and integrating the measurement value is going to yield the amount of consumed fuel. This integration is carried out on-board, and is sent to EDL which displays the consumed fuel. Moreover, the amount of loaded fuel can be set in the EDL before turning on the engine and there is an option to zero the consumed fuel value to start a new measurement. This was done prevously on the EDL ground software, what was sensitive for data gaps in telemetry.

FCC hardware redesign: new RX-MUX unit

Concept

The FCC used during the FLEXOP project had a lot of components which needed improvements to handle the new wing, and provide more functionalities during the FLiPASED test flights. The concept of the FCC have not changed (Raspberry Pi, flightHAT unit and two RX-MUX boards), but the RX-MUX panel gets a major revision. The first version of the FLEXOP FCC's RX-MUX had dsPIC33 MCU-s, and for the redundancy, one RX-MUX PCB board had two microcontrollers on it, working independently. Only the 2S LiPo power supply was common on those panels, all the other power and data lines were separated. The capabilities of the PIC controller became a bottleneck on the FCC, therefore an STM32F4 unit was selected to the new generation of the board. Due to its bigger size (packages have 100 or 144 pins), the MCU redundancy concept had to be dismissed, and only one MCU serves instead of two per PCB. Still, the FCC consists of 2 RX-MUX PCB panels, so redundancy among actuation channels and flight control surfaces still remains in the system.

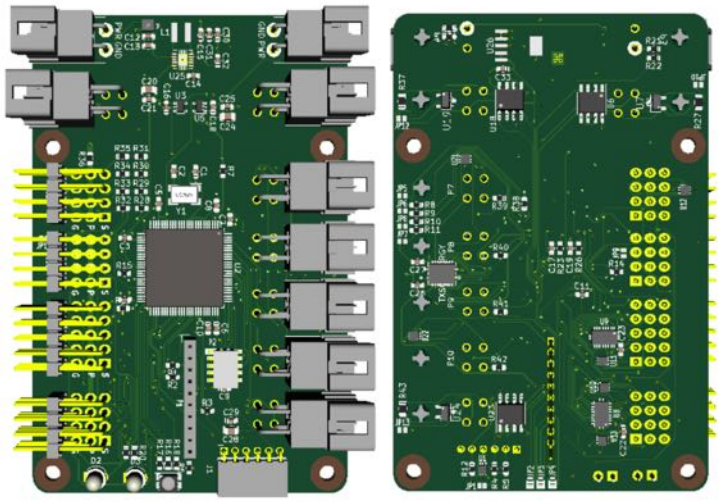


Figure 15: RX-MUX v2 render image

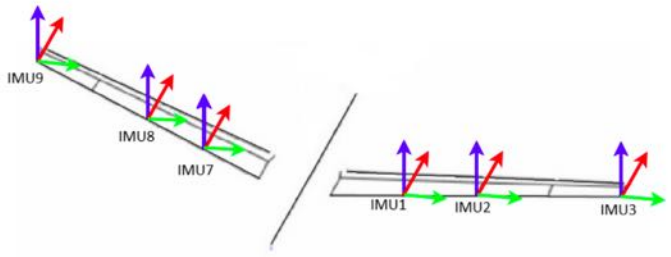
Newly developed features for FLiPASED

- New RC interface for JETI: the analog receiver with PPM signal
- Complete software redesign
- Bootloader / client application development is in progress for easier configuration

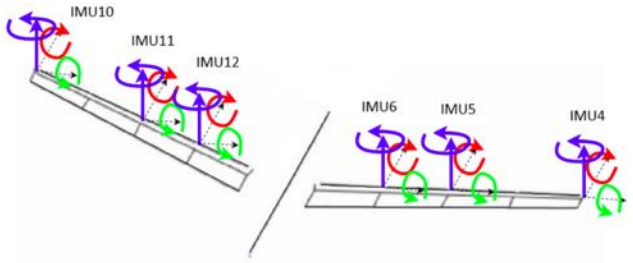
Problems The chip shortage due to COVID-19 pandemic has affected the hardware manufacturing possibilities. Mitigating actions can be: redesign PCB to support both LQFP100 & LQFP144 packages, etc. etc.

New IMUs on the aircraft – tail and fuselage

Based on the lessons learnt during the FLEXOP GVT campaign the onboard inertial sensor instrumentation has been revised to capture fuselage as well as tail and wing in-plane motion. IMU units were sent out by SZTAKI to TUM, where the hardware integration was performed. Based on DLR's suggestion, the aim is to detect different bending modes on the aircraft, so the wing IMUs are not enough anymore. The software integration of these additional sensors started, to log the IMU measurements or send them on the telemetry link.



(a) Leading edge



(b) Trailing edge

Figure 16: IMU configurations

New IMU measurement modes were also developed on IMU units, meaning that the leading edge and the trailing edge units provide different inertial measurements: one mode is dedicated for accelerations, and another is for angular rates. Both modes send Z accelerations for logging. The log parser tool automatically parses the log to have meaningful variable names to simplify log analysis.

DirectDrive integration

The flutter control dedicated high bandwidth Direct Drive actuator has its own controller, FCC gives position commands and gets diagnostic information from the unit via CAN bus. The Direct Drive controller has a state machine, and the communication is based on CAN protocol, using CANopen layer on it, which standardizes some parameter set, query and data type messages. The state machine of the motor controller has to be handled by proper messages, which was the topic of recent development.

Already developed:

- Low level CAN communication
- Initialization after heartbeat messages arrived
- Position commands from RC and from autopilot
- Compatibility solved with other wings with no Direct Drive

Future developments include obtaining diagnostic information and sending it to the GCS via telemetry, robustness improvements of both the software and the hardvare layer of the unit.

OBC-II

To extend the capabilities of the demonstrator aircraft, an additional on-board computer (OBC-II) has been mounted onto the aircraft. The OBC-II is another Raspberry Pi (version 4) which communicates with the existing FCC. The reason behind the addition of the new computer is to reserve the FCC for the execution of critical tasks like the autopilot, logging and telemetry.

The proposed new features to be executed on the OBC-II are for example an online modal analysis tool developed by DLR, additional sensors and 5GHz Wifi telemetry from TUM and a 4G LTE telemetry from SZTAKI.

4G telemetry

The proposed concept of the 4G/LTE telemetry can be seen in figure 25.

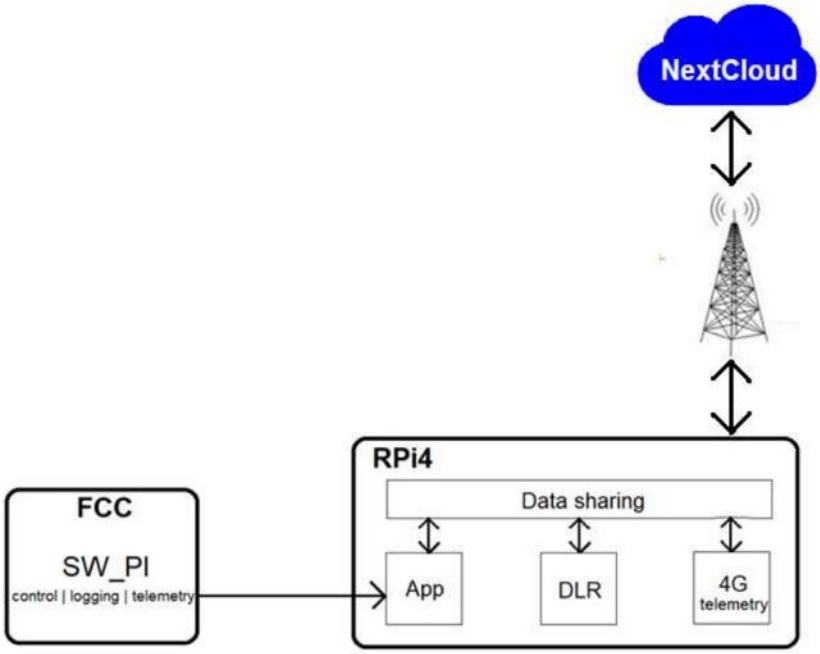


Figure 17: 4G telemetry concept

As it is visible in figure 25, the FCC sends data to the OBC-II via wired Ethernet connection. The ”App” in figure 25 receives the incoming packets and shares it with the other processes running on the OBC-II including the 4G/LTE telemetry. Afterwards, the LTE process transmits the necessary data to a NextCloud server. Of course, the Raspberry Pi 4 does not have a built-in 4G module, but there are several options to choose from. We used a Sixfab Raspberry Pi hat (which can be seen in figure 26 which utilizes a Quectel 4G module to connect to the network.

Figure 18: Sixfab 4G hat for the Raspberry Pi

The current state of the 4G/LTE telemetry is that the upload of small files is feasible, however due to the rather slow uplink speeds large files take too much time to be uploaded. Therefore, the main task for the future is to improve the upload speed of the system to be able to examine data after each flight.

Ground test data

DLR and TUM are responsible of conducting the ground tests and gathering the test data with the airframe. The structural properties of the newly developed wing has to be checked to clear its airworthiness. Before integration of the wing into the demonstrator fuselage a static test campaign will be performed, when sandbags will be placed gradually to the wing to test its structural properties up until 150% of design load conditions. The corresponding load and deformation pairs will be measured and the FEM model of the wing will be updated with built-in tuning beams within the NASTRAN framework.

The teams of DLR and ONERA wil jointly conduct and coordinate the GVT. The Ground Vibration Test (GVT) will take place in collaboration between DLR and ONERA in Göttingen Germany in 2022. A detailed test campaign will see the aircraft suspended from bungee cables and excited in several configurations. The industrial test process of ONERA-DLR will be employed to produce a comprehensive modal model of the aircraft which will be used for Finite Element (FE) model updating, flutter calculations and controller updating. Furthermore investigations into structural non-linearity will also be conducted.

The partners will follow the same test procedure as developed during the FLEXOP project, but several additional improvements will be incorporated. The onboard sensors and the newly developed operational modal analysis routines will be compared with the measurements of the extensive external instrumentation. The -3 wing will incorporate additional number of trailing edge flaps, hence their modal analysis will be also conducted.

Flight test data

TUM is responsible of conducting the flight tests. The teams of TUM, DLR and SZTAKI have devoted significant effort to develop flight test data analysis tools, mostly implemented in Matlab environment.

Test data downloaded from the on-board FCC has to be parsed and checked for errors first. This is done on-site during flights. While the more rigorous quality check is done only after the flights.

Automatic scripts have been developed to speed up the process with repetitive tasks and helping user

centric visualization.

The various command and sensor measurements are translated to meaningful physical values, and compensation post-processing steps are executed to remove outliers and biases in the data: for example the servo position feedback is temperature dependent for which a calibration scheme is applied to reach ± 0.1 deg absolute position error during different flight phases with different altitudes at different velocity ranges leading to temperature variations.

The entire dataset is also split into different parts, to analyse the test points - different engagement points of the autopilot must be analysed in a proper context without additional excess data.

DLR worked mostly on rigid body system identification to recover flight mechanics parameters - with the aim of feeding back those parameters to the overall aircraft dynamical model.

In addition to flight mechanics system identification TUM worked on analysing the flexible effects of the air data boom and internal structure, what shows unwanted oscillatory signals around the estimated accelerations and wind angles. The investigation led to the redesign of the air data boom mounting and IMU mounting points inside the demonstrator instrumentation bay.

SZTAKI developed Matlab based tools to run the actuator compensation automatically with special emphasis on automating the identification of flight phases. In addition to that baseline control performance test are also developed to automatically evaluate the tracking performance of the control loops.

Demonstrator Ground and Flight Test Requirements The deliverable “D1.3 Demonstrator Ground and Flight Test Requirements Definition” lays the foundation for the various testing and corresponding instrumentation tasks in WP3 of the project. In the beginning of the project, several key factors have been identified and objectives as well as performance metrics have been proposed to show the benefits of the MDO tool-chain developed within the project, in correspondence with the demonstrator aircraft. To be able to compare the model predictions with real-life tests a detailed ground and flight test plan has to be developed and the corresponding instrumentation, analysis tools and supporting infrastructure have to be prepared.

This document explains the currently existing legacy ground and flight test equipment and the identified gaps what needs to be developed within the project to fully explore the performance gains proposed by the MDO workflow within FLiPASED. The current demonstrator aircraft needs hardware and software improvements as well as new sensing, telemetry and onboard computing capabilities. The updated instrumentation serves dual role, to better conduct ground test and to execute more relevant flight tests with upgraded functionality.

Demonstrator Status

Demonstrator Airframe The two goals of the legacy FLEXOP project, namely aeroelastic tailoring and active flutter suppression, presented very different requirements for flight tests [114]. Testing of the aeroelastic tailoring would require high load factors on the wings and were not expected to pose any operational challenges. On the other hand, flutter testing demands high airspeeds and, therefore, big areas for manoeuvring (acceleration, deceleration, high-speed turns). Additional requirements were placed by the scale-down task, imposed by industrial partners (geometry similar to a new generation commercial airliner), sensors required for the measurements (minimum 2kg of payload capacity) and limitations due to logistics (maximum part size of the unrigged aircraft should not exceed 4m).

Based on these requirements a flight test mission was designed and, including the UAV design experience of TUM, a preliminary design of the demonstrator was done. This resulted in a 65kg take-off

weight (TOW) demonstrator with a swept, 7m span wing and a V-Tail. The demonstrator received three pairs of wings: the rigid wing for setting the baseline (designated as -0), the wing with active flutter control (-1) and the aeroelastically tailored wing (-2). Risk alleviation by system redundancy was incorporated for aircraft controls. The concept required symmetrical control of the aircraft even if one of the batteries powering the aircraft control surfaces would lose voltage. This requirement resulted in 8 wing flaps (4 per wing) and 4 ruddervators (2 per V-Tail). Additionally as a last measure to protect the infrastructure, a parachute within the tail part of the fuselage was integrated.

Main characteristics of the demonstrator can be found in figure 19. The demonstrator is mostly flown

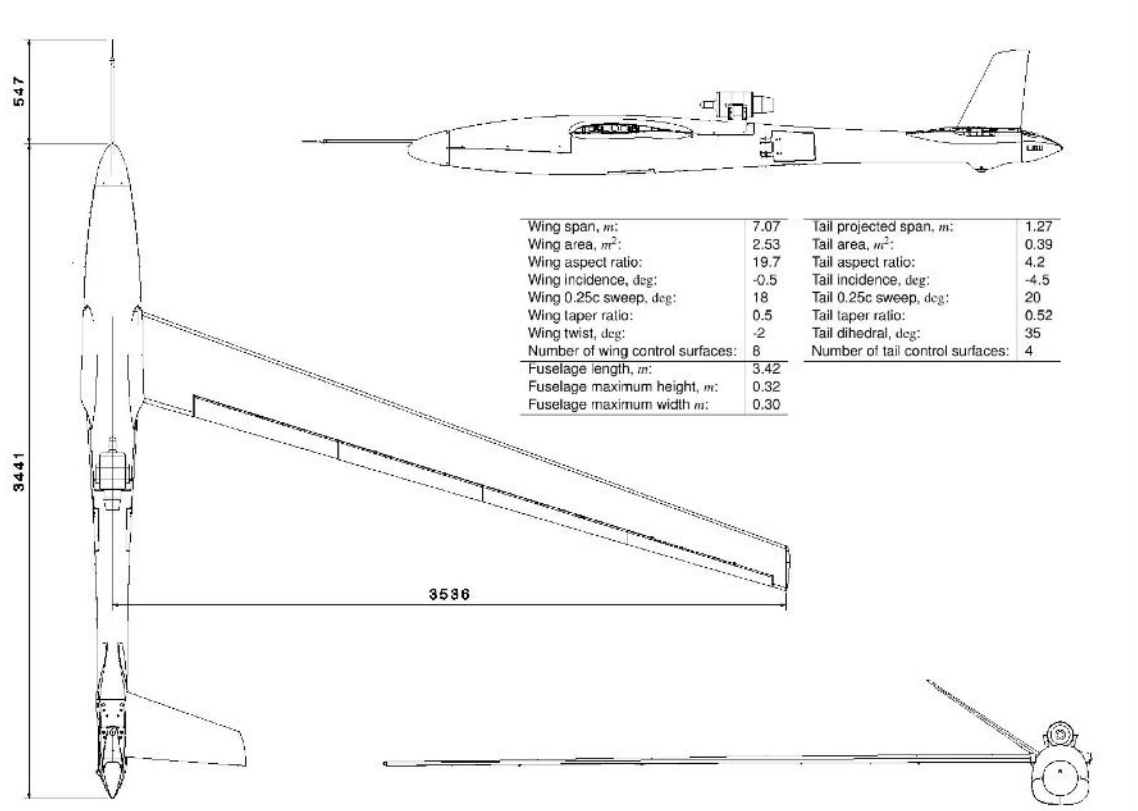


Figure 19: FLEXOP Subscale flight demonstrator. Note that the left wing and V-tail are excluded.

manually by pilot via external vision. Stability augmentation flight mode via angular rate feedback is also available, where in manual mode the surface deflections are directly linked to the joystick positions on the transmitter and the measured angular rates.

The aircraft has two control links. Control via two different transmitter brands was desired to decrease the risk of both transmitters failing together due to a common mode failure (either connectivity, electrical or mechanical). The main one is a Jeti DS-24 system which has an additional back-up receiver that is integrated further away from all the other radio links. The secondary RC system, controlled by the backup pilot, is a Graupner mc-28 system. Graupner has only one receiver with four antennas that are pointed in different directions. In comparison, Jeti receivers have only two antennas.

Propulsion System The main requirements while designing the propulsion system were high acceleration, low vibration and precise speed tracking. Taking these requirements into account, a jet engine

paired with a fast-response airbrake system was selected. The jet engine is a BF B300F turbine with 300N maximum thrust capability. The engine was mounted on a pylon above the fuselage with the fuel tank located directly below it. This was designed with the intent to keep the same centre of gravity throughout the flight.

Other Systems A 5-hole air-data probe provides the measurements of aerodynamic angles and airspeeds, as well as static and total pressures. The measurements are captured within the Micro Air-Data Computer manufactured by Aeroprobe. The probe is mounted on a boom 55cm in front of the demonstrator nose. The boom length was determined using the airflow data received from the Computational Fluid Dynamics (CFD) simulations. During the simulations, airspeed and flow angle values were compared at different distances away from the nose. The distance which resulted in local flow values within 1 percent of the free-stream values was chosen.

A secondary airspeed reading is measured by a low-cost air-data probe mounted on the right V-Tail of the aircraft. To make sure that the readings on the secondary air-data probe are satisfactory for backup operation, the calibration of the probe was checked in the wind tunnel. Furthermore, the airspeed measurements in between the two probes were compared during the first flight of the demonstrator. Good correlation of both measurements gave confidence that even in the case when the main airspeed sensor is lost, a reliable backup would be available.

The position and attitude of the aircraft is measured by a high-precision Inertial Measurement Unit (IMU) MTi 710 manufactured by xSens. Additionally, multiple IMU units were installed in the wings for capturing structural acceleration data during flutter testing. Additionally, fiber Bragg grating (FBG) system was installed in the wings for accurate deflection measurements.

A parachute system, comprising of a drag chute and the main chute, is installed in the aircraft (manufactured by skygraphics AG). In case the chute release is triggered, the magnet, holding the tail cone, is released. The tail cone is then pushed away by the incoming airflow. It has the drag chute attached to it, which, consequently pulls the main parachute out.

Two small cameras (Mobius 1080p HD Action Camera) are integrated within the tail cone. The cameras were placed in a way to overview both wings in-flight and provide visual feedback after test runs. They were not accessible online and would only be used for offline evaluation.

Interior layout of the systems is displayed in figure 20.

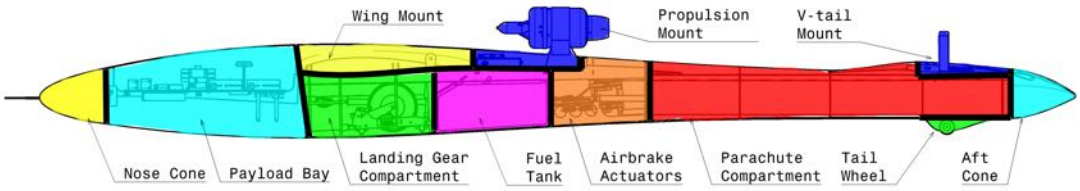


Figure 20: Interior layout inside the demonstrator.

Demonstrator Avionics The avionics system was built from the ground up to serve the custom needs of the demonstrator. No commercial off the shelf system would provide the required number of input interfaces (custom sensors, RC, telemetry, etc.), as well as the output interfaces with 19 PWM servos, and custom UART based propulsion unit interface. The research task of integrating custom, highly sophisticated, modern control and estimation methods, instead of the standard PID gains also facilitated

the need of a custom avionics solution.

Autopilot

Introduction

The flight tests require to test the autopilot features one-by-one, i.e. switching on one loop after another and observe their behavior at the flight test campaign. In order to handle different mission objectives, the pilot should have to command GCS to set the FCC to switch to the desired operation mode. This command is basically sent out from the GCS, and then the pilot decides when to perform it. Therefore, two autopilot levels are distinguished. The Autopilot 1 (AP1) mode is just an augmented mode at most of the time, and in this mode, the state machine inside the autopilot receives the commands from the ground. The Autopilot 2 mode (AP2) performs the preset action, when the pilot decides to switch to it on his 3-level-switch on the JETI transmitter.

Mission Planner interface

The GCS employs a customized Mission Planner which has selectors for the operation modes, what are sent to the FCC via the telemetry antennas. The options displayed by the Mission Planner depend on the capabilities of the software running on-board[2]. Figure 21 illustrates the custom user interface in Mission Planner.

As you can see, there are options to turn on or turn of logging on the FCC and multiple options to alter the functions of the autopilot.

Implementation

The state machine is implemented inside the autopilot's Matlab Simulink model, as a Stateflow Chart, so the code generation from the implemented autopilot logic can be done with the well-known Matlab Simulink Coder. The mask of this state machine in Simulink can be seen on figure 22. The augmented mode is active in all autopilot modes (both AP1 and AP2), and the actually flight tested components are switched on in AP2 mode. This can involve inner loop functionalities, and task specific modes. After a successful test flight with a certain software, the next software's autopilot mode AP1 can include the capabilities which are proven to work in AP2 mode. Then, the flights can be basically performed with switching to AP1 after takeoff, then using AP1 until landing, and switching to AP2 for performing a Mission Planner specified objective. The details of the performed maneuvers are described in the Flight Test Cards.

Another novelty in FLiPASED is that telemetry parameters are logged on the FCC. Only essential parameters were logged previously, what caused the lack of information, so we could not reconstruct the entire chain of events during a test. From the log we could not determine in which state the autopilot was (even though they were logged on ground), which button has been pressed on Mission Planner, and when. Therefore now the parameters which are set by the GCS are logged onboard.

ECU communication, fuel flow display in EDL

The signal of the fuel flow sensor provides a value proportional to the rate of consumed fuel in each measurement step. Thus, scaling and integrating the measurement value is going to yield the amount of consumed fuel. This integration is carried out on-board, and is sent to EDL which displays the consumed fuel. Moreover, the amount of loaded fuel can be set in the EDL before turning on the engine and there is an option to zero the consumed fuel value to start a new measurement. This was done

[2] At a flight test, the flight test crew should be helped by the user interfaces as much as possible. The software on FCC is started with a custom USB pendrive solution, which also selects the autopilot version. This version is reflected at the GCS, therefore only appropriate command messages can be sent out. For example, if a software is dedicated to run engine identification, Mission Planner displays different options than an another software, which is used for injection of modal identification signals. On Figure 21 you can observe the configuration modes and buttons for baseline mode functionality.

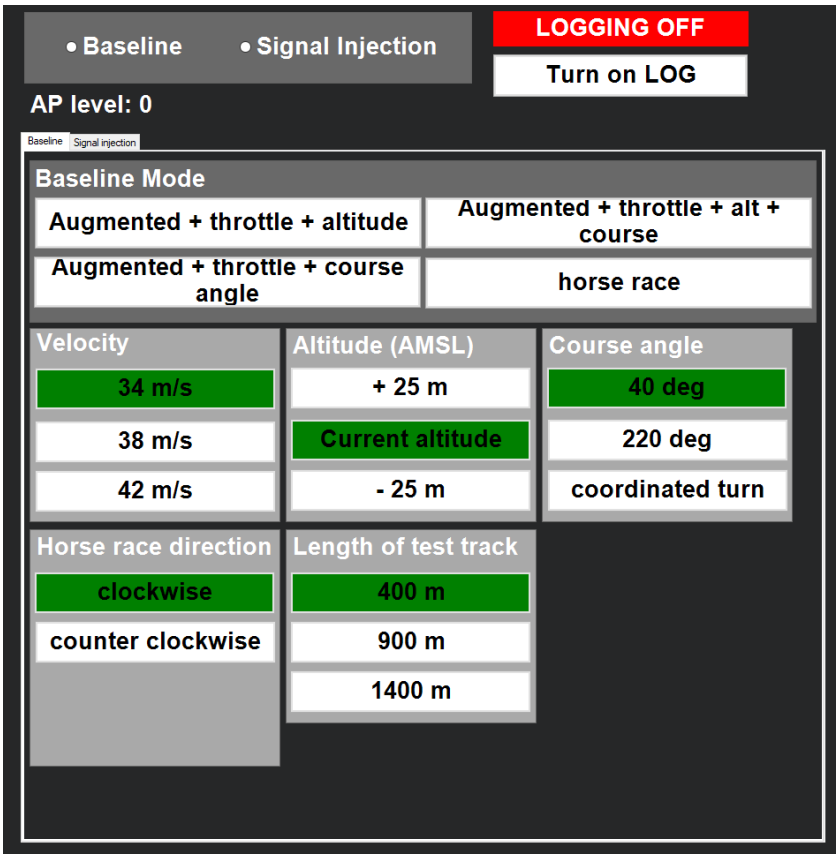


Figure 21: Mission Planner user interface

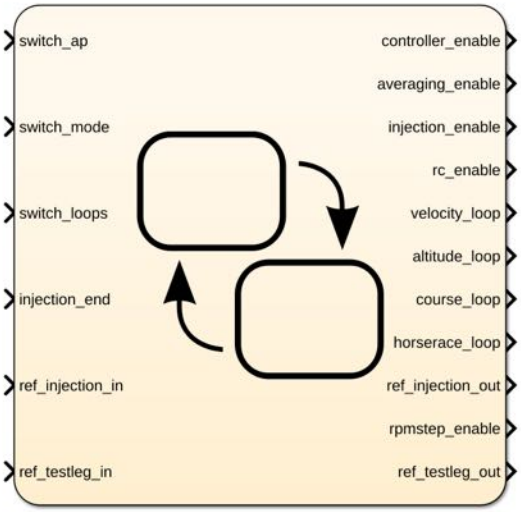


Figure 22: Matlab Simulink StateFlow Chart inside the autopilot

prevously on the EDL ground software, what was sensitive for data gaps in telemetry.

FCC hardware redesign: new RX-MUX unit

Concept

The FCC used during the FLEXOP project had a lot of components which needed improvements to handle the new wing, and provide more functionalities during the FLiPASED test flights. The concept of the FCC have not changed (Raspberry Pi, flightHAT unit and two RX-MUX boards), but the RX-MUX panel gets a major revision. The first version of the FLEXOP FCC's RX-MUX had dsPIC33 MCU-s, and for the redundancy, one RX-MUX PCB board had two microcontrollers on it, working independently. Only the 2S LiPo power supply was common on those panels, all the other power and data lines were separated. The capabilities of the PIC controller became a bottleneck on the FCC, therefore an STM32F4 unit was selected to the new generation of the board. Due to its bigger size (packages have 100 or 144 pins), the MCU redundancy concept had to be dismissed, and only one MCU serves instead of two per PCB. Still, the FCC consists of 2 RX-MUX PCB panels, so redundancy among actuation channels and flight control surfaces still remains in the system.

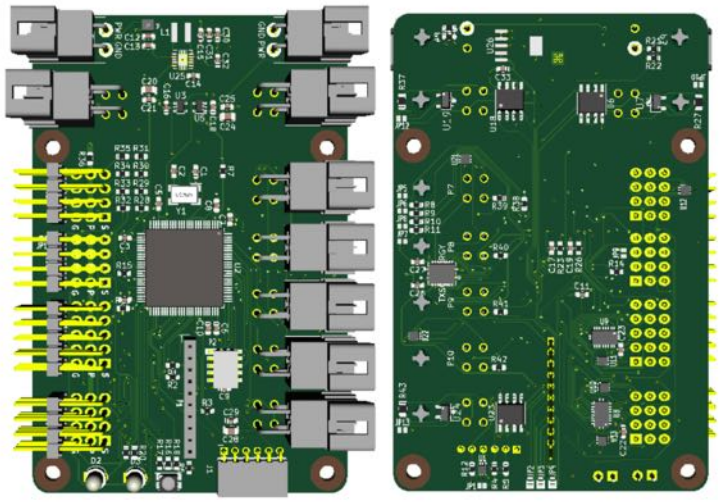


Figure 23: RX-MUX v2 render image

Newly developed features for FLiPASED

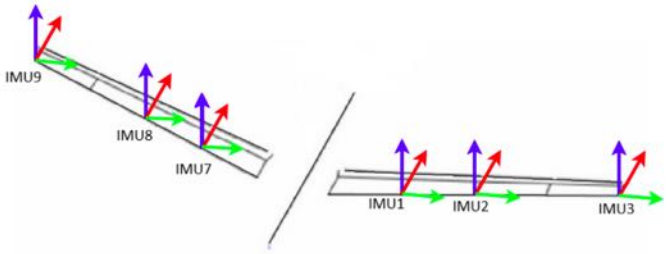
- New RC interface for JETI: the analog receiver with PPM signal
- Complete software redesign
- Bootloader / client application development is in progress for easier configuration

Problems The chip shortage due to COVID-19 pandemic has affected the hardware manufacturing possibilities. Mitigating actions can be: redesign PCB to support both LQFP100 & LQFP144 packages, etc. etc.

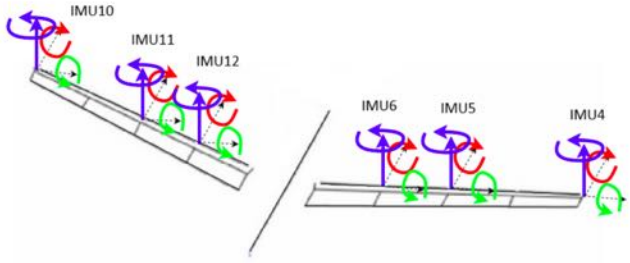
New IMUs on the aircraft – tail and fuselage

Based on the lessons learnt during the FLEXOP GVT campaign the onboard inertial sensor instrumentation has been revised to capture fuselage as well as tail and wing in-plane motion. IMU units were sent out by SZTAKI to TUM, where the hardware integration was performed. Based on DLR's suggestion, the aim is to detect different bending modes on the aircraft, so the wing IMUs are not enough

anymore. The software integration of these additional sensors started, to log the IMU measurements or send them on the telemetry link.



(a) Leading edge



(b) Trailing edge

Figure 24: IMU configurations

New IMU measurement modes were also developed on IMU units, meaning that the leading edge and the trailing edge units provide different inertial measurements: one mode is dedicated for accelerations, and another is for angular rates. Both modes send Z accelerations for logging. The log parser tool automatically parses the log to have meaningful variable names to simplify log analysis.

DirectDrive integration

The flutter control dedicated high bandwidth Direct Drive actuator has its own controller, FCC gives position commands and gets diagnostic information from the unit via CAN bus. The Direct Drive controller has a state machine, and the communication is based on CAN protocol, using CANopen layer on it, which standardizes some parameter set, query and data type messages. The state machine of the motor controller has to be handled by proper messages, which was the topic of recent development.

Already developed:

- Low level CAN communication
- Initialization after heartbeat messages arrived
- Position commands from RC and from autopilot
- Compatibility solved with other wings with no Direct Drive

Future developments include obtaining diagnostic information and sending it to the GCS via telemetry, robustness improvements of both the software and the hardvare layer of the unit.

OBC-II

To extend the capabilities of the demonstrator aircraft, an additional on-board computer (OBC-II) has been mounted onto the aircraft. The OBC-II is another Raspberry Pi (version 4) which communicates with the existing FCC. The reason behind the addition of the new computer is to reserve the FCC for the execution of critical tasks like the autopilot, logging and telemetry.

The proposed new features to be executed on the OBC-II are for example an online modal analysis tool developed by DLR, additional sensors and 5GHz Wifi telemetry from TUM and a 4G LTE telemetry from SZTAKI.

4G telemetry

The proposed concept of the 4G/LTE telemetry can be seen in figure 25.

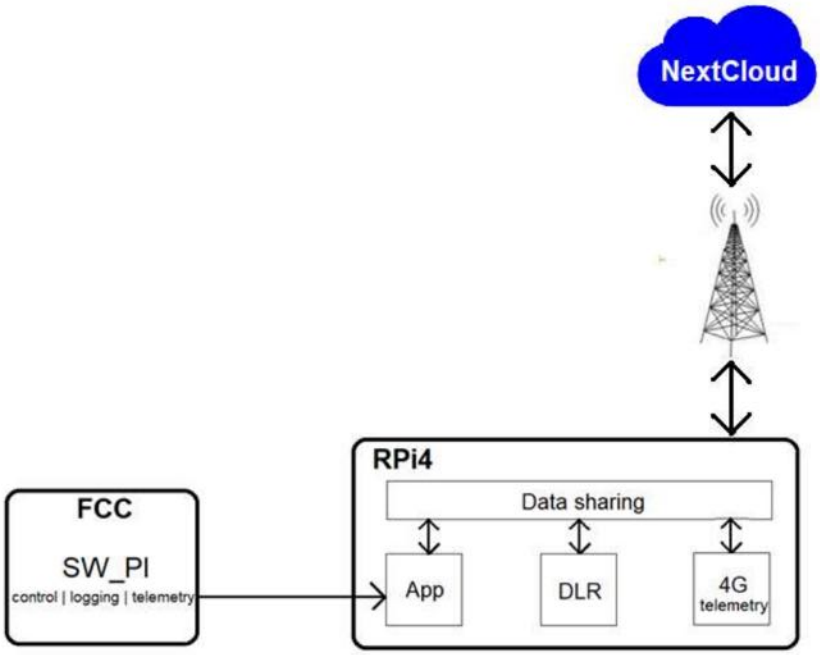


Figure 25: 4G telemetry concept

As it is visible in figure 25, the FCC sends data to the OBC-II via wired Ethernet connection. The ”App” in figure 25 receives the incoming packets and shares it with the other processes running on the OBC-II including the 4G/LTE telemetry. Afterwards, the LTE process transmits the necessary data to a NextCloud server. Of course, the Raspberry Pi 4 does not have a built-in 4G module, but there are several options to choose from. We used a Sixfab Raspberry Pi hat (which can be seen in figure 26 which utilizes a Quectel 4G module to connect to the network.

Figure 26: Sixfab 4G hat for the Raspberry Pi

The current state of the 4G/LTE telemetry is that the upload of small files is feasible, however due to the rather slow uplink speeds large files take too much time to be uploaded. Therefore, the main task for the future is to improve the upload speed of the system to be able to examine data after each flight.

Ground Test Data DLR and TUM are responsible of conducting the ground tests and gathering the test data with the airframe. The structural properties of the newly developed wing has to be checked to clear its airworthiness. Before integration of the wing into the demonstrator fuselage a static test campaign will be performed, when sandbags will be placed gradually to the wing to test its structural properties up until 150% of design load conditions. The corresponding load and deformation pairs will be measured and the FEM model of the wing will be updated with built-in tuning beams within the NASTRAN framework.

The teams of DLR and ONERA wil jointly conduct and coordinate the GVT. The Ground Vibration Test (GVT) will take place in collaboration between DLR and ONERA in Göttingen Germany in 2022. A detailed test campaign will see the aircraft suspended from bungee cables and excited in several configurations. The industrial test process of ONERA-DLR will be employed to produce a comprehensive modal model of the aircraft which will be used for Finite Element (FE) model updating, flutter calculations and controller updating. Furthermore investigations into structural non-linearity will also be conducted.

The partners will follow the same test procedure as developed during the FLEXOP project, but several additional improvements will be incorporated. The onboard sensors and the newly developed operational modal analysis routines will be compared with the measurements of the extensive external instrumentation. The -3 wing will incorporate additional number of trailing edge flaps, hence their modal analysis will be also conducted.

Flight Test Data TUM is responsible of conducting the flight tests. The teams of TUM, DLR and SZTAKI have devoted significant effort to develop flight test data analysis tools, mostly implemented in Matlab environment.

Test data downloaded from the on-board FCC has to be parsed and checked for errors first. This is done on-site during flights. While the more rigorous quality check is done only after the flights.

Automatic scripts have been developed to speed up the process with repetitive tasks and helping user centric visualization.

The various command and sensor measurements are translated to meaningful physical values, and compensation post-processing steps are executed to remove outliers and biases in the data: for example the servo position feedback is temperature dependent for which a calibration scheme is applied to reach ± 0.1 deg absolute position error during different flight phases with different altitudes at different velocity ranges leading to temperature variations.

The entire dataset is also split into different parts, to analyse the test points - different engagement points of the autopilot must be analysed in a proper context without additional excess data.

DLR worked mostly on rigid body system identification to recover flight mechanics parameters - with the aim of feeding back those parameters to the overall aircraft dynamical model.

In addition to flight mechanics system identification TUM worked on analysing the flexible effects of the air data boom and internal structure, what shows unwanted oscillatory signals around the estimated accelerations and wind angles. The investigation led to the redesign of the air data boom mounting and IMU mounting points inside the demonstrator instrumentation bay.

SZTAKI developed Matlab based tools to run the actuator compensation automatically with special emphasis on automating the identification of flight phases. In addition to that baseline control performance test are also developed to automatically evaluate the tracking performance of the control loops.

Ground Test Requirements

Taxi tests for landing gear and handling Building on previous flight test experiences, landing gear proved to be one of the biggest challenges during the operation of the demonstrator. The aircraft was very difficult to control while on the ground, leading to a few very dangerous situations and one accident, where the aircraft skidded of the runway and hit a runway light. Therefore, upgrades were necessary to ensure sustainable operation of the aircraft.

Parallel with mechanical upgrades, computer simulations were made to help identify further problems without risking the air-frame itself discussed in section 2.1.3.

As a starting point, the following landing gear design flaws have been identified:

1. Very narrow main landing gear makes it easy for the aircraft to bank from wingtip to wingtip. If this happens during takeoff or landing, the wingtip touches the ground and instantly creates a destabilizing moment.
2. Main landing gear is longitudinally far from the center of gravity. This means that the disturbing bank angle, required to tip the aircraft, is further decreased.
3. The tires of the main landing gear are too soft for the airplane. This makes it possible to deform the tires very easily and also significantly increases the rolling resistance during take-off run.
4. Unsteerable tail wheel makes the aircraft very hard to control while on the ground. The tail has to be lifted up first and aircraft is then steered with the rudder.

Two different concepts for fixing the landing gear were discussed:

1. Fundamentally changing the landing gear layout.

2. Adjusting the current landing gear to make it acceptably safe for operation.

Because of the fact that the first option would require major fuselage changes and would take at least a few months, it was decided to start with the second option first. Ways to improve handling were discussed during the winter before the first flight test campaign. Due to the complex nature of the problem the solutions that were initially agreed upon did not completely resolve the issue. This resulted in an iterative process with different concepts being implemented as add-ons to the initial design along the way. The chronology of the process was:

1. Implement the steerable tailwheel with damping.

   The initial solution to steering was to install an off-the-shelf tailwheel assembly. Unfortunately, the solution did not work because the load on the tailwheel appeared to be too big for the part. Therefore another, completely custom iteration was done. This included a custom milled aluminum fork for steering and a damping assembly. The damping assembly was composed of glass-fiber-reinforced plastic plate acting as a leaf spring for longitudinal damping and two rubber dampers for lateral stiffness. The structure held well, but the steering made the aircraft hard to control and very sensitive to any pilot inputs.

2. Change the brakes of the main landing gear to more effective ones.

   Tire brakes were changed to drum brakes. From previous testing it was noted that the tires wear out very quickly due to the brakes. Also, the braking power of the old system proved to be too little. Therefore, new type of brakes was implemented that would both conserve the tires and increase the braking force on the wheel hub.

3. Add a gyro to the tailwheel.

   Introducing the steerable tailwheel did not solve the controlability problem as the team has hopped. The aircraft became very sensitive, especially at higher speeds. The solution was to introduce a gyroscope-based compensation for the gain on the steering. This proved to improve the steering somewhat.

4. Reverse the main landing gear frame to shift the ground contact point back.

   One of the main findings, mentioned in the early research on taildragger aircraft is that the tendency to veer of the runway is decreased if the centre of gravity is kept as close as possible to the main landing gear. This was recorded in all the reports on the topic. Therefore, changing the location of the landing gear was considered. Luckily, the landing gear frame was easy to flip, moving the main landing gear backwards by 75mm. The outcome was lesser tendency to veer off the runway, an increase to the critical bank angle to tip on one wing, but also higher load on the main tires. Even though the weight increase was only 2.5% per wheel, the main tires were already overloaded before. The further steps would include looking for stiffer main tires, if possible.

5. Laterally stiffen the main landing gear assembly

   During the taxi tests cameras were mounted facing both the gears. This helped to observe the behavior of the landing gear and make further conclusions. One of them was that the main landing gear is too flexible laterally, which makes it easier to tip onto one wing and harder to get out of the tipped position. Therefore, further parts were introduced to stiffen the landing gear laterally.

6. Change the main wheels to stiffer ones

   Even though the gear was made stiffer, it was recognized that the tyres of the main gear are way too soft for the aircraft. This was discovered during one of the testing days, where the aircraft stood on the ground for a couple of hours. As a result the foam-filled tyres deformed plastically and were not usable anymore. Additionally, during high speed taxi tests a set of tyres burst into

pieces after they got too hot (Due to braking and rolling). It was decided that a stiffer tyre is a must. And with no alternative tyres available for the same wheelset, a double sailplane tailwheel (TOST 150 MINI) instead of the original RC model grade wheels were bought. The TOST wheels would have a proper inflatable tyre moutned on, which would make the main gear stiffer laterally.

7. Add brakes with higher efficiency

   In addition to upgrading the wheels to stiffer ones, the TOST wheels also had a possibility to have disc brakes mounted on them. Since long braking path was also discovered to be a problem during our flight tests, this seemed like a good option.

The changes of both, main gear and tailwheel resulted in a considerably more steerable aircraft. Multiple taxi tests were done, including low speed and high speed tests, to make sure the aircraft has enough controllability to safely resume flight testing. In the end, changing the main wheels from RC model grade to aviation grade seemed to make the biggest difference. The aircraft was declared as flight-worthy again. The main requirement was adequate pilot feel, what is difficult to quantify, but during the iterative retrofit solution the pilots have gained insight on the boundaries of ground handling envelope and had a clear and consistent go/no go decision threshold after each taxi test.

Analysis and simulation results The goal of creating a simulation framework was to be able to analyze the ground handling behaviour of the Aircraft with different structural and layout parameters. That allowed to test different physical configurations and develop trends based on them, which would normally have needed risky taxi test potentially leading to permanent structural damage of the airframe.

Type	Condition	Improvement
Inherent stability of configuration	unstable	toe-out main wheels, Configuration change
Lateral (yaw) Stability	stable	
Rollover Stability	critical	Increase of $V_{ro,cr}$, change of T/O flap configuration
Tip-over Stability	very good	
Chance of veering off/ground loop	high	fix tail wheel, increase lateral friction of tail wheel

Table 1: Summary of T-FLEX stability analysis

Table 1 shows the main results of the ground handling stability analysis of the T-Flex demonstrator. Due the tail-wheel or tail-dragger configuration, the ground behaviour itself is inherently unstable. Any side-force experienced by the vehicle will result in a destabilising moment. Making the vehicle stable, would require a conceptual landing gear change to either a tricycle or a quadricycle configuration.

The rollover stability of the vehicle can be improved by varying the parameters in equation 1 in a way, that the $V_{ro,cr}$ is higher than the take-off speed. As of current state, none of the indicated parameters can be changed to increase $V_{ro,cr}$ without permanent structural change of the air-frame itself.

$$V_{ro,cr} = \sqrt{\frac{gRbl_2}{2hL}} \tag{1}$$

To decrease the chance of ground side-skid during operation the lateral friction must be increased if possible, by forcing the tail down using the elevator in more upward setting during taxi, take-off and landing operations.

The results of the simulations showed, that there is a considerably high difference in dynamic response of the vehicle, at the speed when the tail-wheel lifts off from the ground. With the current configuration, that switch occurs around 17-18 $\frac{m}{s}$, which is well below the normally experienced takeoff speed 30-33 $\frac{m}{s}$. At the point, when the tail-wheel has no contact with the ground, the vehicle instability drastically increases while the pilot can use only the low effectiveness rudder input as counter acting control surface.

To make the operation of the T-FLEX safe and reliable, the take-off and landing problem has to be mitigated. The taxi test results as well as the simulation results showed, that the original landing-gear design of the vehicle is not sufficient for the task at hand. With the small configuration changes, the system is still not reliable enough to allow us to honestly say, "It will survive the takeoff."

Possible solutions which worth to consider are the followings.

- decrease take-off speed, so the in-stable/uncontrollable phase should be minimized
- Landing-gear design change
  - Non-retractable tail-wheel configuration
  - tricycle configuration, either retractable or not
- design and build of a take-off cart

Given the remaining time-frame of the project and the complexity of the development/deployment and overall testing of a new landing-gear design, the take-off speed redcution is favourable.

Decreasing take-off speed Investigations of different flap configurations have been carried out. TUM built a 50% scaled demonstrator of the demonstrator called "Defstar" with which they tested the stall behavior of the aircraft and also investigated the effects of various flap settings. After a number of confidence building stall recovery maneuvers and the investigation of stall behavior of the "Defstar" vehicle, it was decided that increased flap settings during takeoff and landing would not result in dangerous stall behaviour while it would lead to higher lift at lower speeds and the takeoff length could be shortened. The numerciacal predictions and flight test results have been confirmed in a flight test when a decrease of the take-off length was confirmed, leading to significantly better ground handling behaviour. The process of aerodynamic investigation and test results are already described in detail in D3.2 Flight Test Report - Phase 1.

Hardware-in-the-loop (HIL) testing Due to the significant effort of conducting flight tests the main method to clear any newly developed system component or software function is to test it in the HIL simulation platform. It has two distinct versions, one is hosted on a legacy Windows 10 based PC, running Simulink Real Time, and inerfaced with external devices via the standard PCI cards of a desktop PC. This system has two instances: one at SZTAKI (for software development) and another one at TUM (for pilot training). The other HIL is based on a Speedgoat target machine, which is a turnkey solution with dedicated hardware interfaces and dedicated software implementation of the required communication protocols between the simulation and hardware components, this is also available at two locations (SZTAKI uses it for SW/HW development and another one is under commission at DLR to develop the necessary real-time capable simulation platform for V and V).

Testing autopilot functionality One key activity is developing improved versions of the autopilot for the demonstrator at SZTAKI in Hungary and testing them before flights in Germany.

- signal injection signals, amplitudes, LUT, time before injected signal
- autothrottle, throttle injection
- inner loops, course angle, altitude hold

Remote testing during pandemic Due to restrictions during the pandemic work in the office was very limited. Hence a custom solution was implemented to support remote work during COVID-19 pandemic. The objective was to perform full HIL tests remotely:

- Access to computers at the lab
- Accessing the FCC
- Emulate transmitter behavior for FCC
- Programming of the microcontrollers and power supplies

The aforementioned tasks were completed successfully and HIL tests were successfully performed from home.

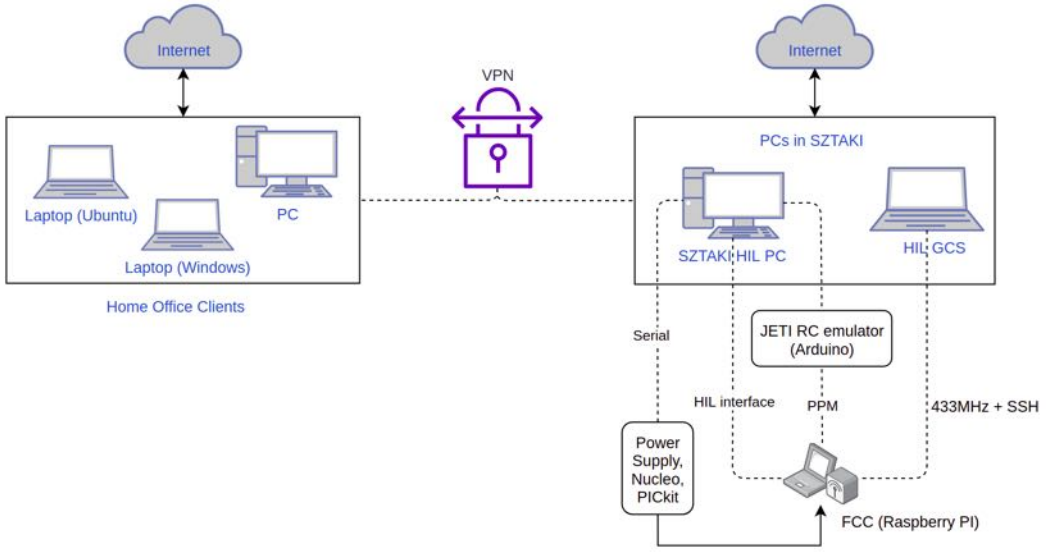
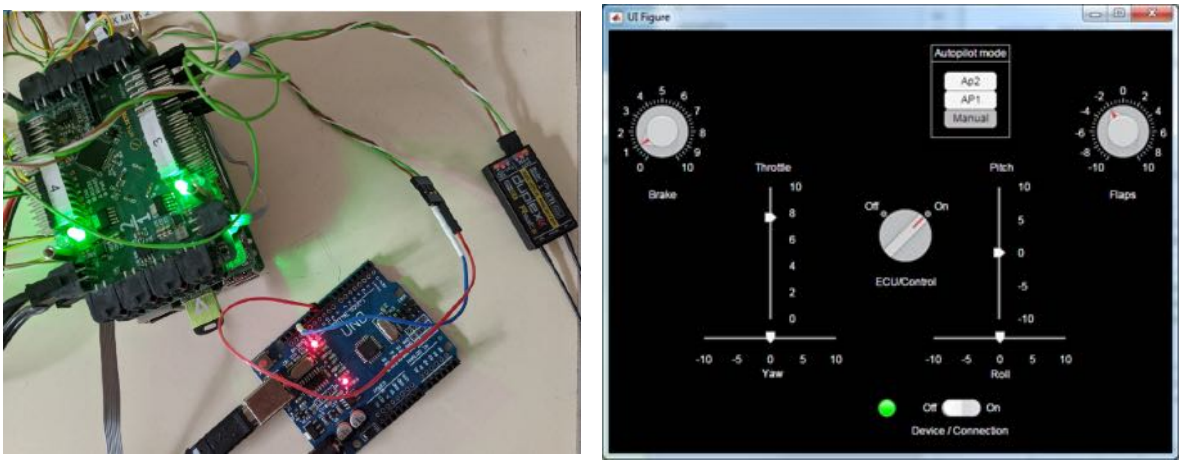


Figure 27: Remote configuration of SZTAKI HIL for home office access

- VPN and VNC access was configured: Linux and Windows clients are both supported
- FCC was accessible with WiFi sticks from multiple PCs at the lab, and SSH connection could be established for start and stop the software, and copy logs
- Programmable power supply units were connected to HIL PC and was controlled by serial port messages
- Arduino application was developed for PPM generation to emulate JETI transmitter behavior, running parallel with the JETI receiver as seen on Figure 28a.

- A client application in MATLAB was developed with JETI interfaces (Autopilot switch, joysticks), as seen on Figure 28b.
- Command-line interface is also provided in MATLAB



(a) Arduino PPM generator device

(b) Matlab GUI for PPM generation

Figure 28: Remote HIL testing equipment

Pilot Training Pilot Training Simulator (PaOT) is necessary for the flight test team to get familiar with the aircraft dynamics, the autopilot capabilities and the GCS user interface. The whole mission have to be practiced with as close to the real mission as possible. Therefore, all interface have to work (almost) the same, as the real aircraft.

Autopilot testing has reached the point, when GCS issues a lot of commands on the Mission Planner interface, as seen on Figure 21. These buttons are controlling a state machine in the software (Figure 22). Basically the FLEXOP PaOT ran a SIL simulation on a target machine, connected to a visualization environment, and running the aircraft model provided by DLR. The telemetry functionalities included emulating MAVLink messages, but only the general ones, not those parameters which are set for autopilot parametrization. EDL was also not implemented in this simulator.

- The statemachine of the autopilot is implemented in MATLAB Stateflow, but unfortunately at TUM PaOT there were no license for this. Therefore it could not work. Also, putting inside a generated S-function is a wrong solution, because Autopilot is a referenced model, and S-functions can't deal with it.
- There were no MAVLink emulation in the project before, to implement almost the whole protocol, which is already in C code, into MATLAB, would be too much effort for this issue. However there are other ways, such as a HIL method, using the FCC itself for the MAVLink communication.
- Another problem is the different architecture and compiler, therefore mex files cannot be shared among the computers, so deployment is complicated, because development in Simulink is not enough, but a new software requires to build an autopilot software running on Raspberry, and a different mex binary from the same autopilot to run in pilot training environment.

These problems resulted that in the current form, the original PaOT would need a lot of development to achieve the same functionallity as the SZTAKI HIL. So we concluded that we replace the former

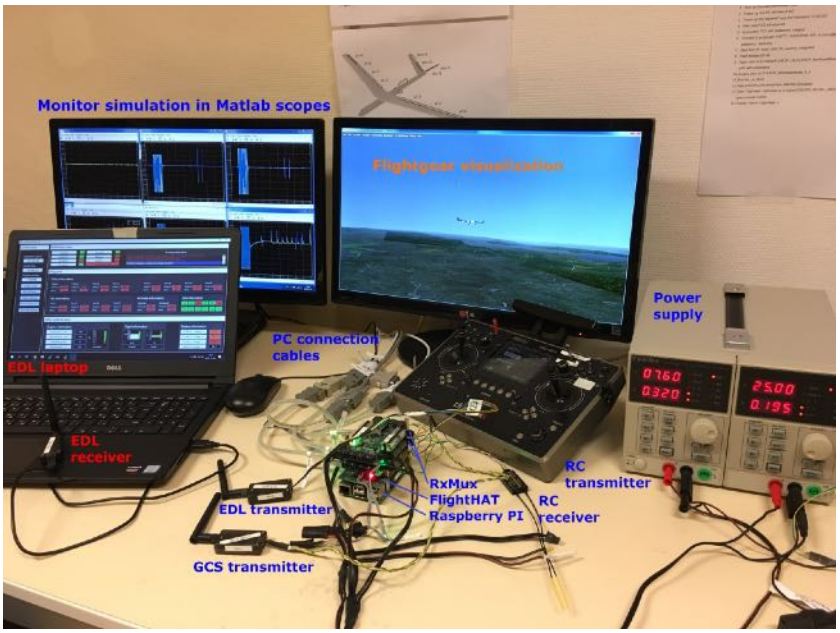


Figure 29: SZTAKI HIL which was sent to Munich

PaOT SIL, to the SZTAKI HIL, to have the same environment at SZTAKI for development and testing before software releases and at TUM for pilot training before the flight testing of the released software. Therefore, the same software can be tested in SZTAKI, at TUM, and flown actually.

The integration of new features to SZTAKI HIL is reaching its limits soon, so this setup will be easier to maintain in Speedgoat environment. That environment will serve as a common, universal platform with industrial grade hardware.

Speedgoat Integration Implementing the aircraft model on the Speedgoat machine is in progress. The input capture module successfully reads the PWM signals coming from the RX-MUX units of the FCC. Therefore, now the actuator signals can be received through the CAN interface and directly from the PWM input capture units as well. Snippets from the CAN and PWM input capture Simulink blocks can be seen in figures/D103 30 and 31.

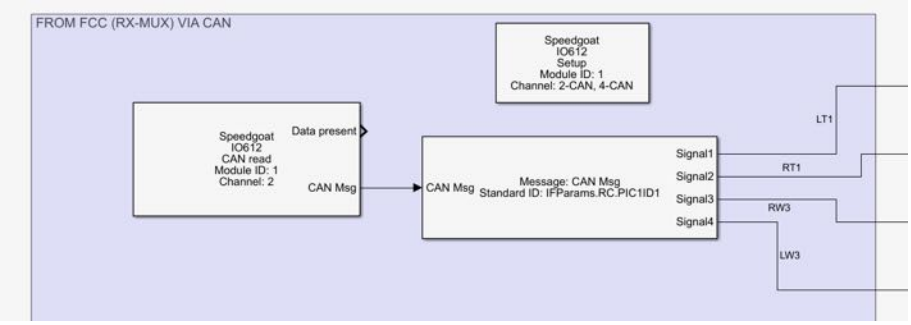


Figure 30: CAN input block of the Speedgoat Simulink model

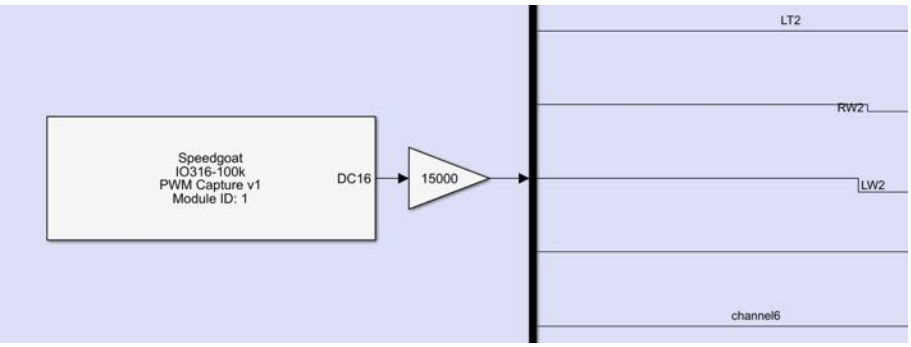


Figure 31: PWM input capture block of the Speedgoat Simulink model

Unfortunately, the S-function aircraft model from DLR was not compatible with the Speedgoat simulation out of the box, however the bottom-up model from SZTAKI provided realistic aircraft dynamics.

The development of the sensor emulators are still under development. The emulated air data sensor sends data correctly, but the xSens model needs some modifications to make it work.

Therefore, the upcoming tasks are to finish the sensor interfaces to provide data for the FCC and to integrate the DLR aircraft model to the Speedgoat simulation.

Ground Vibration Test For a successful Ground Vibration Test (GVT) the aircraft should be transported and assembled in the DLR test facility in Göttingen. During testing the Flight Control Computer (FCC) should also manage the actuator control. All equipment will be provided by DLR and ONERA. Close collaboration between the partners who use the data for model updating will also be beneficial for the project. Finally, if the modifications to the empennage and fuselage are considered significant, it will be necessary to re-test all wing sets.

In the continuation, the collection of the measurements should be stored in a place accessible to all partners to perform analysis and extraction of the modal content and reconstruct a digital version of the experiment.

Engine Thrust Measurement TUM developed and tested an engine thrust measurement system.

In order to quantify the effect of active drag reduction, as will be done within Task 2.5 Tool Adaptation: Control Design (and partially tasks Task 2.2: Tool Adaptation: Aerodynamics, Task 2.3: Tool Adaptation: Aeroelasticity and Task 2.4: Tool Adaptation: Movables Design), accurate measurement of changes in drag will be necessary. For manned aviation this is usually done by glide polar method (for sailplanes) or by calculating the thrust applied together with aircraft acceleration measurements (for powered aircraft). In the latter case, thrust of the engine is usually provided by the engine manufacturer for specific flight conditions and is later adapted by measuring engine parameters (temperatures, pressures and revolutions).

In case for T-FLEX demonstrator (or in fact most of the UAVs), only very limited engine data is provided by the manufacturer. Usually, fuel consumption, idle and maximum thrust and RPMs can be found. But thrust data, required for estimating drag of the aircraft, is not available. Therefore, it was decided to measure the thrust of the engine directly in-flight.

In order to detect changes due to active drag reduction, the measurement system accuracy has to be of the same order of magnitude as the difference in drag. During preliminary stages, it was estimated that absolute reduction of 0.5N could be expected. Moreover, the system has to last the whole flight

(around 30 minutes), be able to withstand maximum trust of the engine (300N) and have a sufficiently high sampling rate for such application. As the jet-engine in use is relatively slow response, 50Hz sampling rate was decided to be enough in comparison with 200Hz used otherwise on the aircraft.

Environmental conditions also had to be taken into account. Temperature, altitude and pressure as well as weather-induced conditions such as wind and rain were to be expected. Additionally compensation for off-level flight conditions was to be possible. Measurement of net thrust was required.

Table 2 summarizes the requirements.

Table 2: Summary of design requirements for the thrust measurement system

Sub-Requirement	**Value**
Range of Measurement	$0 \leq T \leq 300$ N
Precision of Measurement	± 0.5 N
Duration of Measurement	≥ 30 *minutes*
Sample Rate	≥ 50 *samples per second*

Flight Test Requirements

System identification The online system identification methods (operational modal analysis) developed by DLR will be used during the flight test campaign. This system receives data from the Flight Control Computer (FCC) performs signal processing, modal analysis, and tracking and sends the results via telemetry to the ground station. Here engineers can monitor critical damping trends as an indicator of flutter onset. As the system matures during testing, a connection to the controller providing real time state matrices could be further investigated.

There are three critical components what are required for these methods to work, what were not implemented on the demonstrator before the start of the project:

- A secondary, non-flight critical on-board computer running the algorithms,
- Change in the sensor configuration on the wings and additional new IMUs inside the tail surfaces,
- Reliable telemetry channel and GCS user interface to monitor the behaviour of the system.

All three items have been resolved and the demonstrator is ready to perform tests with the system on-board.

Baseline control The key components of the baseline controller have been laid down in the FLEXOP project, however the new challenges necessitate further adjustments.

As depicted in Figure 32 the architecture of the controller has been selected to be structured, in order to facilitate sequential testing and validation. This control architecture also allows the possibility of reconfiguration by introducing additional loops, as discussed later.

The successful testing of the inner loop functionalities (namely: pitch attitude and lateral directional control) have been performed. According to the feedback from the pilots and the flight test crew, minor adjustments, additional tunings have also been applied on the control loops. In order to test the full functionality of the baseline controller and validate the model-based design, Figure 33 summarizes the proposed flight test plan schedule.

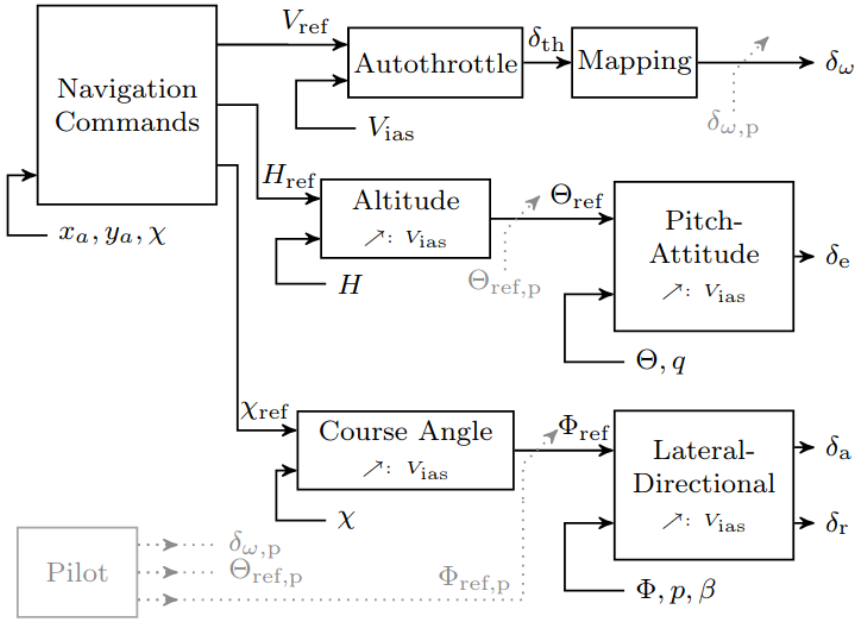


Figure 32: Architecture of the baseline controller

Before each flight test, the implemented autopilot software goes through a series of ground tests in order to check the basic behaviour of the control loops. These ground tests involve the imitation of certain maneuvers with fixed airspeed and by checking the deflection of the control surfaces in response of the maneuvers.

The satisfactory performance of the inner loop functionalities allows stable straight and level flight conditions to be achieved, where additional signals can be injected for identification purposes. Accordingly, the baseline control architecture has been extended with the functionality of injecting test and identification signals superimposed on the stabilizing inner loops.

A crucial component of the hardware configuration is the BF300 jet engine and the corresponding autothrottle control loop. Due to the lack of experimental data about the engine's behaviour, along with some unmeasured signals, a model-based look-up table has been created to describe the non-linear response of the engine. This non-linearity is included in the baseline control architecture (see Figure 32) and flight no. 6. is dedicated for the validation of this. It is not possible to directly assess the engine parameters, therefore a reverse engineering approach has to be applied, comparing the flight measurements with the ones predicted by the high fidelity model. Accordingly, a prediction-error method can be applied to determine the unknown (or uncertain) parameters. Once the engine parameters are adjusted the further functionalities of the baseline controller can be tested.

One aspect of the baseline controller flight testing is the sequentiality: the separate functions can be tested separately in various flight test scenarios. Flight tests no. 8 and 9 are dedicated for the outer-loops and the way-point tracking functionalities and consist multiple tests.

An important and crucial point of the baseline control flight testing is the feedback it provides for the model-based design methodology. Namely, the measured flight data has to be evaluated and compared with the response of the model-based toolchain (see Figure 34). These measurements, along with the expertise of the flight test crew, are essential for the fine tuning of the control loops. In addition they can

No.	Wing	Title	General
1	0.2	Taxi Test 1	Assessment of ground handling qualities.
2	0.2	Maiden Flight 1	Assessment of In-Flight Behaviour of Systems and Handling Qualities when flown by external pilot. Manual flight control only.
3	0.2	Maiden Flight 2	Public Maiden Flight
4	0.2	Air-Data Probe Calibration 1	Airspeed and altitude sensor calibration.
5	0.2	Flight Mechanics Test 1	Flight mechanics model identification. Doublets and step inputs on roll/pitch/yaw.
6	0.2	System Test 1	Engine model identification.
7	0.2	System Test 2	Airbrake model identification. Fly manoeuvers required to calibrate the airbrake model (low negative pitch manoeuvres with extended airbrakes)
8	0.2	Autopilot Test 1	Assessment of autopilot functionality and autonomous flight. Autopilot inner loop and course angle hold tests. Mode switching, altitude hold, IAS hold, WPS tracking.
9	0.2	Autopilot Test 2	Assessment of autopilot functionality and autonomous flight. WPS tracking including speed and altitude changes in between.
10	0.2	Autopilot Test 3	Check if the autopilot can hold a steady load factor (n_z) during turn. Check if the autopilot can follow the horse track closely.
11	0.2	Envelope Expansion 1	Turns with increasing bank angle (increasing load factor)
13	0.2	Systems Test 3	Testing of the direct drive. Perform full direct drive frequency sweep to identify its' influence on flight dynamics and aeroelastic modes.
14	0.2	Aeroelastic Test 1	Aeroelastic model identification. Sine sweeps on control surfaces. Multiple repetitions.
15	0.2	Flutter Test 1	Open loop flutter test. Flying one test leg, download data, verify that the speed can be increased further on, increase the speed for the next test leg. No flutter control.

Figure 33: Flight test plan related to the baseline controller [89]

provide valuable insights on the modeling and design methodology, formulated as formal metrics and incorporated in the integrated design.

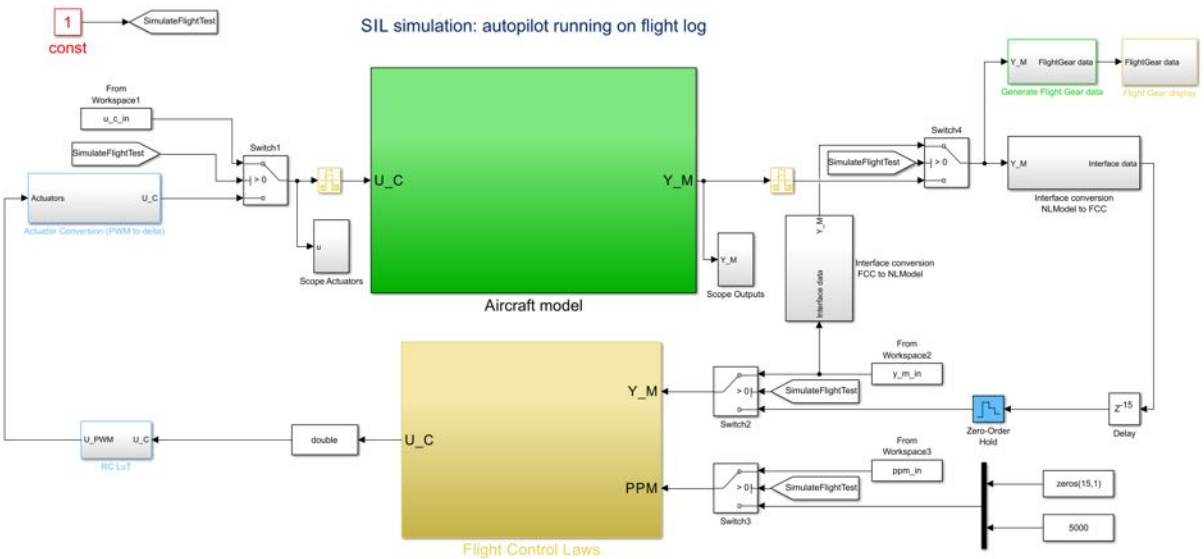


Figure 34: SIL simulation of the baseline controller based on flight log measurements and inputs

Load control The functions are developed jointly by the DLR-ONERA team, as a novelty for the FLiPASED demonstrator. Both maneuver and gust loads control loops will be assessed in flight (not in ground). The common objectives of these control functions is to limit the worst case loads in either maneuver or gust episodes. In addition, a common constrain is to maintain the baseline flight performances unchanged (or slightly unchanged).

For the both MLA and GLA, the outer ailerons are used together with IMU sensors. The synthesis of the control functions is made automatically based on a single model, obtained after approximation and leads to a single LTI MIMO control law.

To address the performance of these control laws the structural response of the wing will be measured and loads and deformation will be estimated either via visual-inertial or fibre brag based measurements. Additional strain sensors might be placed on the root section of the wingspars. The quantification of these load alleviation functions also require precise flight dynamics and air data reconstruction to be able to compare gust to response amplitudes with load alleviation functions turned on and off during various external weather conditions.

Flutter control The nominal flutter controller is developed by SZTAKI. The flutter controller [109] aims to mitigate the undamped oscillations of the wings that occur if the aircraft is flying beyond the flutter speed. It uses the the outermost aileron pair to achieve this goal. For the control design, two uncertain models of the aircraft are constructed: one captures the longitudinal behaviour (hence the symmetric flutter mode), and the other the lateral behaviour (hence the asymmetric flutter mode). The airspeed, and the frequency and damping of one of the structural modes are considered uncertain. Also, dynamic uncertainty is included to account for dynamics neglected because of the model order reduction. Two SISO controllers are designed using the two models. The objective of the design is to minimize the sensitivity function of the closed-loop while limiting the bandwidth of the controller to prevent the excitation of high-frequency dynamics. The two SISO controllers are blended together to obtain the final MIMO controller and implemented inside the aircraft FCC.

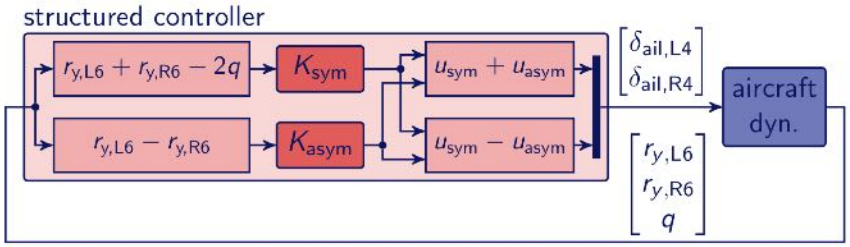


Figure 35: The structure of the closed loop with the flutter controller

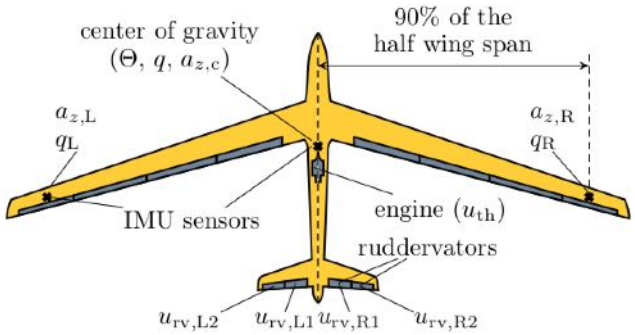


Figure 36: Sensors and actuators of the aircraft

The main requirement to test these controllers are divided into three branches:

- Flight safety related: the wing is equipped with flutter tuning masses and the flight test campaign with flutter (-1) wings will commence without flutter tuning masses, resulting in significantly higher flutter speed. The pilots and flight test team has to gain experience with the softer wings and also with the custom direct-drive actuators, before clearing the vehicle to conduct actual flutter tests.
- Graduality related: the functionality of the flutter control laws will be tested first on the ground to provide structural damping, before flight test could start. Later during the flight test the flights would not exceed $40\frac{m}{s}$ which is far from the open-loop flutter speed. At these speeds the theoretically predicted open-loop vs. closed-loop structural damping values have to match the ones estimated from the flight test results before the flight envelope can be extended to go closer to $50\frac{m}{s}$, the open-loop flutter onset speed. The flight patter is divided into test legs, where the velocity is increased in $2\frac{m}{s}$ increments, and the corresponding damping trends are analyzed before the next speed is commanded.
- Performance related: The demonstrator requires very precise velocity tracking to make sure it does not exceed the target airspeed by $1 - 2\frac{m}{s}$. This is especially important since the vehicle conducts the turns with lower speed and accelerates to the target velocity in straight test legs, with limited space (due to visual line of sigt requirements). For this purpose, and due to flight safety at EDMO airfield, it is a crucial requirement that the vehicle is able to track the target airspeed with sufficient precision.

Baseline performance for comparison From the six flight performed within FLEXOP project, an initial performance picture of the T-FLEX demonstrator could already be assembled. However, no performance identification data was gathered due to mostly unstable flight environment resulting in high scatter of data.

To be able to compare the performance gains realised within the project FLiPASED, a more detailed baseline needs to be set. This baseline will be set using the ”stiff” wing.

The performance baseline should include:

- Setting up a drag baseline with the help of the thrust measurement system (see section 2.1.3).
- Getting more information about the take-off performance of the aircraft in order to optimise take-off procedures.
- Investigate measurement repeatability of different sensors.

The team devoted significant effort in making the flights more deterministic and to reduce the spread of measurement data by developing common procedures and implementing automated test instead of hand flown maneuvers. To establish a better baseline the following procedures have been implemented:

- 3D laser scanning the entire fuselage with the nominal wing and empanage incidence angles, and carefully adjusting these angles before every flight test,
- Calibration and fine adjustment of the aircraft weight and c.g. location before every flight,
- Implementing a maneuver injection function into the autopilot, which stabilizes the aircraft at the corresponding test trim point and injects a fully repeatable time sequence onto the flight control surfaces. Eliminating the imperfections caused by pilots hand trimming the aircraft and flying system identification maneuvers manually.

With the above mentioned improvements the team have collected and evaluated several flights and found very good match between consecutive flights and mathematical model based predictions, hence establishing the baseline performance is on good track.

Benefit and toolchain prediction evaluation ONERA is responsible for the seamless integration and interaction of these different flight control functions. Each modelling step and control function is constructed in a cascaded manner to address a dedicated objective (flight, load and flutter or load prediction). Therefore, attention should be paid to the actual effects when interconnecting of all these functions. This interaction is central in the control function development (almost as much as the performance itself) and should be handled by the proposed toolchain. It is also central in when considering the manner the models are constructed and the assumptions performed.

The benefit and toolchain prediction evaluation claims to engage metrics in accordance to the sought objectives. As an illustration, one may consider the load alleviation, flutter speed, modal content accuracy resolution. This can be done during the ground and flight test experimentation. Indeed, the comparison of the toolchain metrics with the one obtained in experimental campaign will hep to adjust the steps of the process. Figure 37 illustrated the toolchain steps. Each box is a function that shall be evaluated and rated during either the ground test (for example FEM model), or during the flight test (for example peak gust response with GLA on and off).

The basic considerations to execute the required ground and flight tests with the improved demonstrator have been laid down in the present document. To be able to show the benefits of the improved wing and the corresponding design framework several tests have to be executed with the currently existing wingset to provide the baseline performance figures. This necessitates the need to instrument the aircraft with new sensors and improved avionics. Some of the ground tests and laboratory tests have to be repeated with the improved demonstrator. This is followed by ground and later flight testing of the

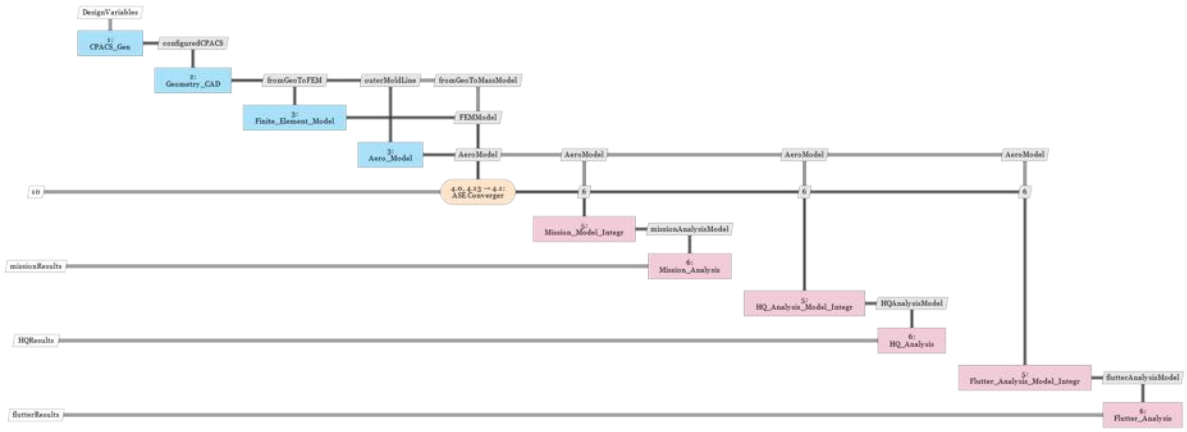


Figure 37: Toolchain steps illustration.

new wingset - which has improved performance and different structural and aerodynamic performance. Hence, ground calibration, GVT testing as well as pilot operational training and flight tests with increasing complexity has to be executed relying on the improved avionics and data analysis tools developed within the project.

Information and interfaces definition for Collaborative Work Process The deliverable ”D1.4 Information and interfaces definition for Collaborative Work Process” lays the foundation of the MDO toolchain developed in WP2 of the project, in correspondence with the demonstrator aircraft. In the beginning of the project, several key factors have been identified and objectives as well as performance metrics have been proposed to show the benefits of the MDO tool-chain. The interdisciplinary teams within the project share models, data and tools among them. D1.4 formalizes these steps within the iteration loops and establishes a document to define their interdependency and their standard interfaces (CAD, NASTRAN, Dymola, Matlab/Simulink, python, embedded C code). In addition to the MDO toolchain, an interface to the HIL tests needs to be defined as well. The HIL tests serve as crucial investigation on whether the developed models and controllers are implementable on hardware as well as their final assessment before the flight tests. Finally, the developed tools in the MDO toolchain are evaluated in flight tests as well after the HIL tests. The lead beneficiary for the deliverable is SZTAKI, but all other consortium partners TUM, ONERA,and DLR contributed significantly to the deliverable by various aspects of the interface definitions and data sharing definitions and MDO toolchain setup tests.

Interdisciplinary Design Architectures and Status There are three main toolchains developed within FLiPASED, which can be seen in Figure 172.

The main toolchain is the MDO toolchain that starts from the CPACS aircraft geometry definition and finishes with the developed aircraft geometry/parameter set tightly coupled with the baseline, manoeuvre load alleviation, gust load alleviation and flutter suppression controllers evaluated in the mission analysis. Based on a successful mission analysis the controllers and the models need to be handed over to the HIL test block. The HIL model needs to be real-time executable and the controllers in discrete from. The HIL tests asses the performance of the controllers in addition to the implementation aspects. The third main block are the flight tests. This block serves as the final maturity test of the developed methodologies. Conclusion are drawn from all three blocks which are then feed back to the CPACS model generation step.

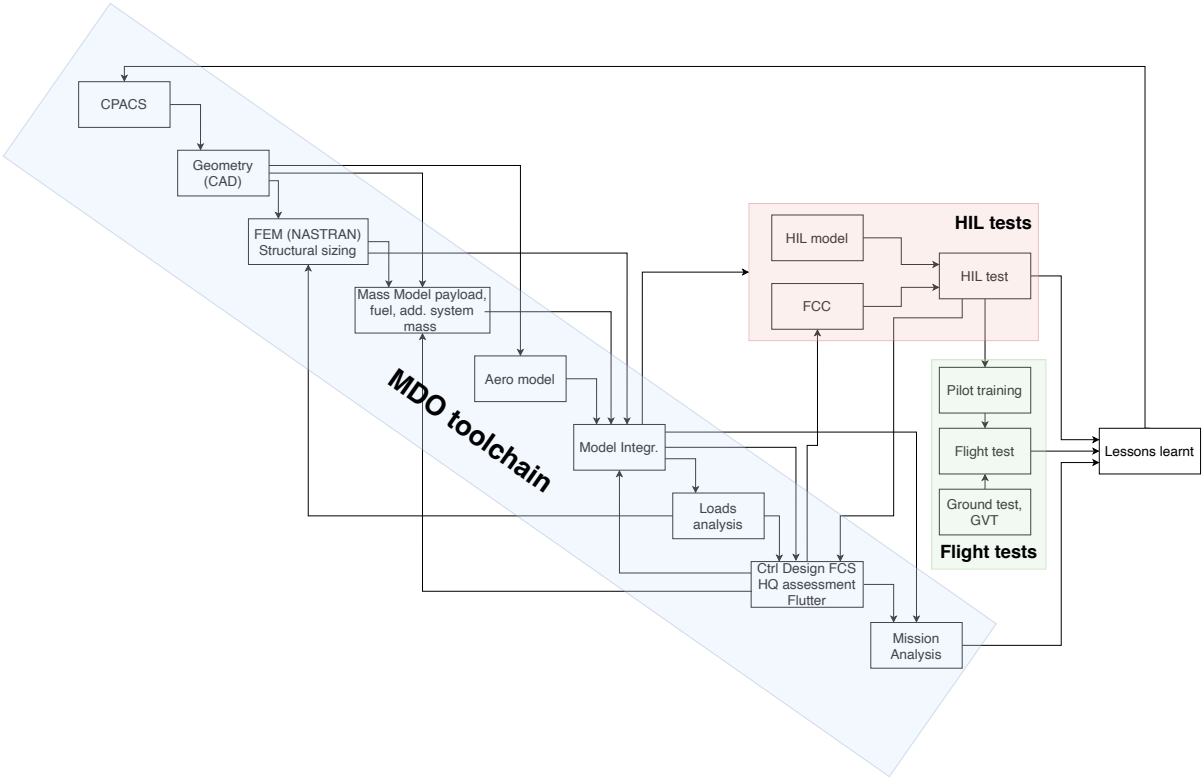


Figure 38: Toolchains developed in FLiPASED

MDO toolchain development status The MDO toolchain development is carried out in two parallel branches. In the first branch each partner implements its own set of tools into the RCE framework locally. This step requires capturing the requirements of the input data coming from the previous MDO block and defining the output data the actual block is creating. This branch of the MDO toolchain is mostly finalized and only minor adjustments are needed.

In the second branch, the communication between the partner blocks and data sharing is implemented. In this case first a simple toolchain is set up for creating variables that are then shared among the partners. The example workflow is shown in Figure 39. This test was carried out successfully. As the final step, the full blocks of the first branch need to be set up to communicate between various partners and to be able to share data. This step is currently under ongoing development and in the next FLiPASED meeting (22/11/2021) at TUM the implementation aspects will be verified and the initial MDO toolchain run is expected to be carried out.

Hardware-in-the-loop (HIL) testing status Due to the significant effort of conducting flight tests the main method to clear any newly developed system component or software function is to test it in the HIL simulation platform. It has two distinct versions, one is hosted on a legacy Windows 10 based PC, running Simulink Real Time, and inerfaced with external devices via the standard PCI cards of a desktop PC. This system has two instances: one at SZTAKI (for software development) and another one at TUM (for pilot training). The other HIL is based on a Speedgoat target machine, which is a turnkey solution with dedicated hardware interfaces and dedicated software implementation of the required communication protocols between the simulation and hardware components, this is also available at two locations (SZTAKI uses it for SW/HW development and another one is under commission at DLR

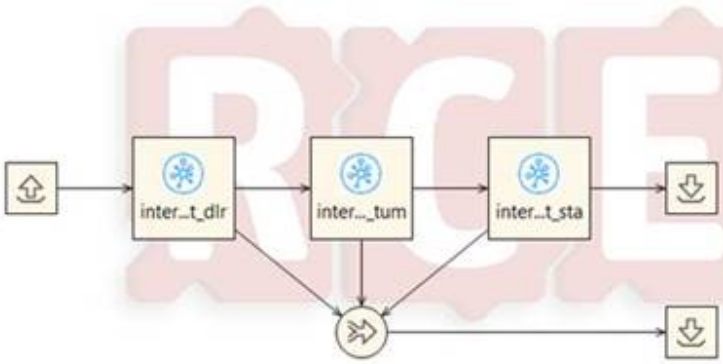


Figure 39: RCE workflow for the testing the communication and data sharing between partners

to develop the necessary real-time capable simulation platform for V&V).

The inputs (controllers and models), input and output interfaces have been defined to the HIL tests. The only remaining items are the discrete versions of the MLA and GLA controllers, what are under fine-tuning, to be able to test the complete ASE system in the HIL environment.

Flight testing status The flight test with the demonstrator aircraft are running in parallel with the MDO toolchain development. First the operation of the aircraft is investigated, then system identification tests are carried out and finally the maturity of the developed controllers within the MDO toolchain will be evaluated. The detailed scope and test schedule of these flight test campaigns are discussed in D3.1, D3.2, D3.3, D3.6, D3.8 and 3.11 respectively.

Aircraft Geometry and FEM Model generation Since 2005 DLR develops the Common Parametric Aircraft Configuration Schema, short CPACS. It contains a parametric description of aircraft configurations as well as the complete transport system, e.g. fleet and airport descriptions. On the other hand control system related layout and parameter information is not standardized in it.

The number of interfaces in multi-disciplinary aircraft design is crucial for a flexible and efficient flow of information. Along with CPACS the number of interfaces between analysis modules is not only reduced but also do these become replaceable, as all adapt to one common definition.

The CPACS format allows the automatic generation, validation and documentation of data-sets. As a part of the Release Kit, CPACS format, documentation and sample configurations are made available at https://www.cpacs.de/ or at https://github.com/DLR-SL/CPACS.

The Common Parametric Aircraft Configuration Schema (CPACS) is a data definition for the air transportation system. CPACS enables engineers to exchange information between their tools. It is therefore a driver for multi-disciplinary and multi-fidelity design in distributed environments. CPACS describes the characteristics of aircraft, rotorcraft, engines, climate impact, fleets and mission in a structured, hierarchical manner. Not only product but also process information is stored in CPACS. The process information helps in setting up workflows for analysis modules. Due to the fact that CPACS follows a central model approach, the number of interfaces is reduced to a minimum.

CPACS Generation CPACS Generation block is the first block in the MDO toolchain. It will generate a CPACS file as shown in Figure 40 which holds aircraft configuration and optimisation variables, for

instance, airfoil, planform, structure layout and so on. All these information and optimisation variables needs to be given as the input for this block. A Matlab script is used to initialize the CPACS data.

Node	Content
?-? xml	version="1.0" encoding="utf-8"
cpacs	
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation	CPACS_Schema.xsd
header	
vehicles	
materials	
aircraft	
model	
uID	FLEXOP
name	FLEXOP
reference	
fuselages	
wings	
wing	
uID	WR
symmetry	x-z-plane
name	WR
description	WR
parentUID	FU
transformation	
sections	
positionings	
segments	
componentSegments	
wing	
analyses	
profiles	
wingAirfoils	
fuselageProfiles	
toolspecific	

Figure 40: CPACS data

Geometry Model Updating Geometry Model Updating block takes the CPACS data as the input. Tixi (https://github.com/DLR-SC/tixi) and Tigl (https://github.com/DLR-SC/tigl) libraries are used to extract geometry and structure information from CPACS. A Catia macro is used to update the existing wing model to the latest parameters. A Catia model will be the only output of this block, as shown in Figure 41.

FEM Model Generation FEM Model Generation block takes the updated Catia model as input and uses TCL programming language in HyperMesh to generate a FEM model. All relevant meshing parameters, modelling techniques and interfaces with fuselage model and empennage model are predefined in macro script of HyperMesh. A Nastran wing model is generated with an established numbering scheme and outputted to several bdf files in a folder with predefined structure, as shown in Figure 42.

Aerodynamic Model Generation Aerodynamic Model Generation takes the CPACS file as input and extracts airfoil and wing planform using Tixi and Tigl library. The TiGL library uses the OpenCASCADE CAD kernel to represent the airplane geometry by NURBS surfaces. The library provides external interfaces for C, C++, Python, Java, MATLAB, and FORTRAN. A Python script is used to generate DLM panel model and written out to bdf files as predefined file structures as shown in Figure 438

RCE Integration All aforementioned blocks are integrated into RCE as shown in Figure 44. RCE is a distributed integration environment for scientists and engineers to analyze, optimize, and design complex systems like aircraft, ships, or satellites. It is especially suited for multidisciplinary collaboration.

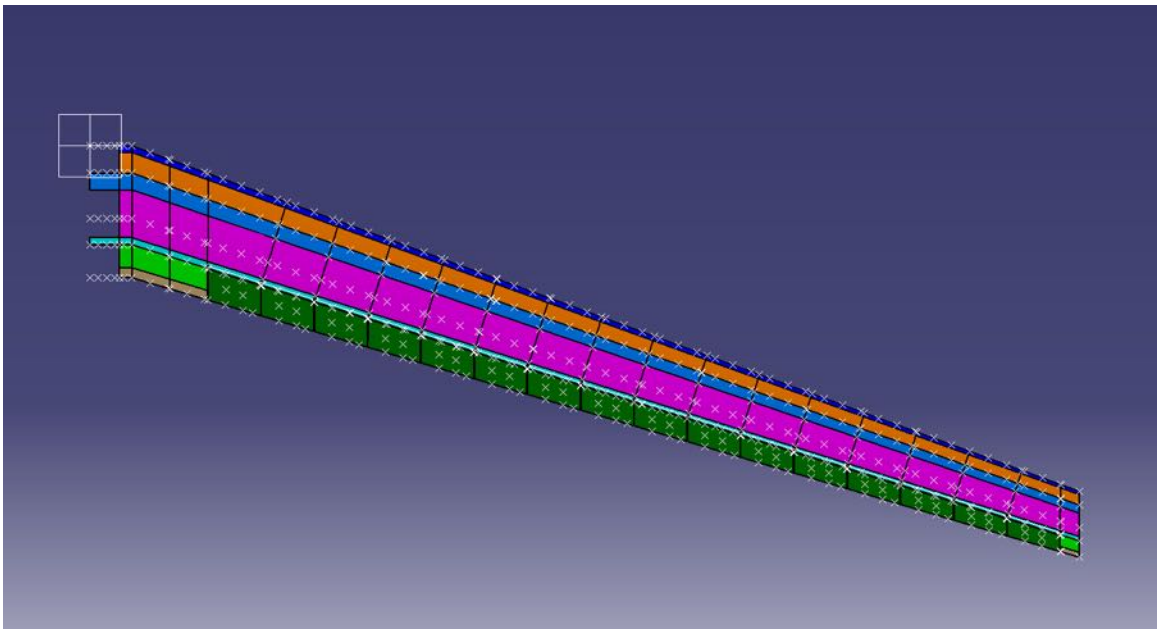


Figure 41: Catia wing model

rbe2_LW.nastran	07.10.2021 10:12	NASTRAN-Datei	8 KB
rbe2_MW.nastran	07.10.2021 10:12	NASTRAN-Datei	4 KB
rbe2_RW.nastran	07.10.2021 10:12	NASTRAN-Datei	8 KB
rbe3_LW_NoUM.nastran	07.10.2021 10:12	NASTRAN-Datei	10 KB
rbe3_MW_NoUM.nastran	07.10.2021 10:12	NASTRAN-Datei	4 KB
rbe3_RW_NoUM.nastran	07.10.2021 10:12	NASTRAN-Datei	10 KB
rbe2_LF.nastran	07.10.2021 10:12	NASTRAN-Datei	6 KB
rbe2_RF.nastran	07.10.2021 10:12	NASTRAN-Datei	6 KB
rbe3_LF_NoUM.nastran	07.10.2021 10:12	NASTRAN-Datei	7 KB
G_rbe2_LW.nastran	07.10.2021 10:12	NASTRAN-Datei	13 KB
G_rbe2_RW.nastran	07.10.2021 10:12	NASTRAN-Datei	13 KB
rbe3_RF_NoUM.nastran	07.10.2021 10:12	NASTRAN-Datei	7 KB
G_rbe2_MW.nastran	07.10.2021 10:12	NASTRAN-Datei	4 KB
G_rbe2_RF.nastran	07.10.2021 10:12	NASTRAN-Datei	9 KB
G_rbe2_LF.nastran	07.10.2021 10:12	NASTRAN-Datei	9 KB
set1.nastran	07.10.2021 10:12	NASTRAN-Datei	5 KB
set1_RW.nastran	07.10.2021 10:12	NASTRAN-Datei	4 KB
set1_LE.nastran	07.10.2021 10:12	NASTRAN-Datei	5 KB
set1_LW.nastran	07.10.2021 10:12	NASTRAN-Datei	4 KB
set1_MW.nastran	07.10.2021 10:12	NASTRAN-Datei	4 KB
set1_LF.nastran	07.10.2021 10:12	NASTRAN-Datei	4 KB
set1_TE.nastran	07.10.2021 10:12	NASTRAN-Datei	5 KB
csm.nastran	07.10.2021 10:12	NASTRAN-Datei	1.656 KB
set1_RF.nastran	07.10.2021 10:12	NASTRAN-Datei	4 KB
rbe3_LW.nastran	07.10.2021 10:12	NASTRAN-Datei	14 KB
rbe3_LF.nastran	07.10.2021 10:12	NASTRAN-Datei	9 KB
rbe3_MW.nastran	07.10.2021 10:12	NASTRAN-Datei	4 KB
rbe3_RW.nastran	07.10.2021 10:12	NASTRAN-Datei	14 KB
csm_right.nastran	07.10.2021 10:12	NASTRAN-Datei	882 KB
rbe3_RF.nastran	07.10.2021 10:12	NASTRAN-Datei	9 KB

Figure 42: Defined file architecture for wing FEM model

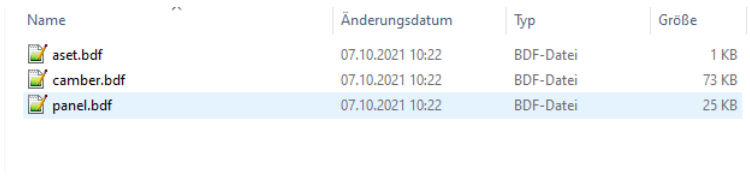


Figure 43: Defined file architecture for wing DLM model

Handling complex systems requires many experts and several tools for analysis, design, and simulation. Using RCE, these tools can be shared between team members and integrated into automated, executable workflows. RCE is extensible and supports different scientific applications with a wide range of requirements.

Corresponding wrappers are written in python to enable the integration. An additional Tigl viewer block is added to the workflow to visualize the aircraft configuration.

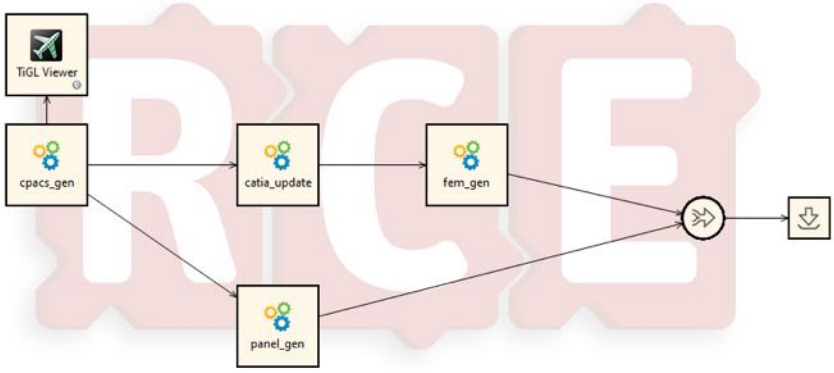


Figure 44: Integrated blocks in RCE

Aeroelastic Model Generation

Aeroelastic Model Integration - NASTRAN From the perspective of the RCE workflow, the input to the NASTRAN aeroelastic model generation block are the following.

1. *CPACS.xml* - containing the most recent aircraft CPACS dataset
2. *wingFE* directory - directory containing the FE and DLM models of the wing, generated by TUM
3. $principal\_angle\_shifts_{1,2}$ float variables - outer-level optimization variables that define the prinicipal angle with respect to which the laminates in the upper and lower skin are oriented

The wing models are generated by the preceding block following an established numbering scheme for the entire aircraft, together with defined interfaces for assembly with the fuselage and empennage

models. This ensures that different configurations of the wing model are compatible with the existing fuselage and empennage models, generated based on FLEXOP data. The input wing model to this RCE block has a defined file-folder hierarchy as shown in Figure 45.

Figure 45: Defined folder-file architecture for wing models from TUM

The NASTRAN aeroelastic model integration block primarily performs the following tasks.

1. Create a modified wing FE model by rotating the existing laminate definitions on the upper and lower skins according to the input variables $principal\_angle\_shifts_{1,2}$
2. Assemble the aerodynamic model of the aircraft by merging the panel definitions, spline sets and the camber correction entries for the wing, fuselage and empennage
3. Run pre-defined NASTRAN decks corresponding to modal, aeroelastic, flutter analyses and a static Guyan reduction
4. Aggregate the output data, including mass and stiffness matrices, and pre-defined aerodynamic bulk data into the output directory

The outputs from this block include two directories and the CPACS dataset as shown in Figure 46.

1. *51-nastran-data* directory - contains the outputs required by the next partner in the RCE workflow, DLR-SR in this case. Files include the mass and stiffness matrices, aerodynamic bulk data - panel definition and camber correction, and other outputs needed for the tools downstream.
2. *51-flipased-ac* directory - contains different NASTRAN solution decks for various analyses in order to aid in debugging.
3. output CPACS dataset - for the demonstrator workflow, the CPACS dataset is not altered during the execution of the tool. For the scale-up workflow, information from analyses such as structural weight, thickness and material properties of the various structural entities can be appended.

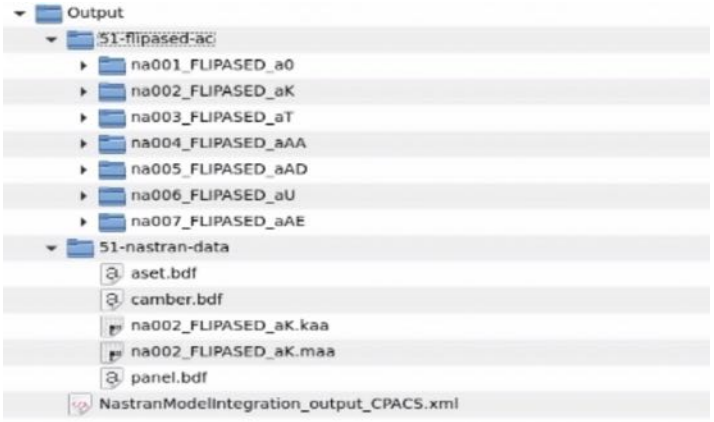


Figure 46: Defined folder-file architecture of NASTRAN aeroelastic models to DLR-SR

Aeroelastic Model Generation and Simulation The aeroelastic model generation and simulation workflow implemented in RCE is given in Figure 47. The workflow is executed from the left to the right. All the corresponding functions are executed in Matlab. The result of each individual block is saved in a Matlab struct. First the aerodynamic, structural and spline grid information as well as mass and stiffness matrices are provided to the first block called ”varloads model”. VarLoads is a tool created in Matlab for defining flexible aircraft models by e.g. setting-up aerodynamic influence coefficient matrices and performing an eigenvalue analysis of the aircraft structure. The results are passed on to the block ”create model input”. The data is then downsized and provided in a specific form, so it can be used with the Simulink simulation environment.

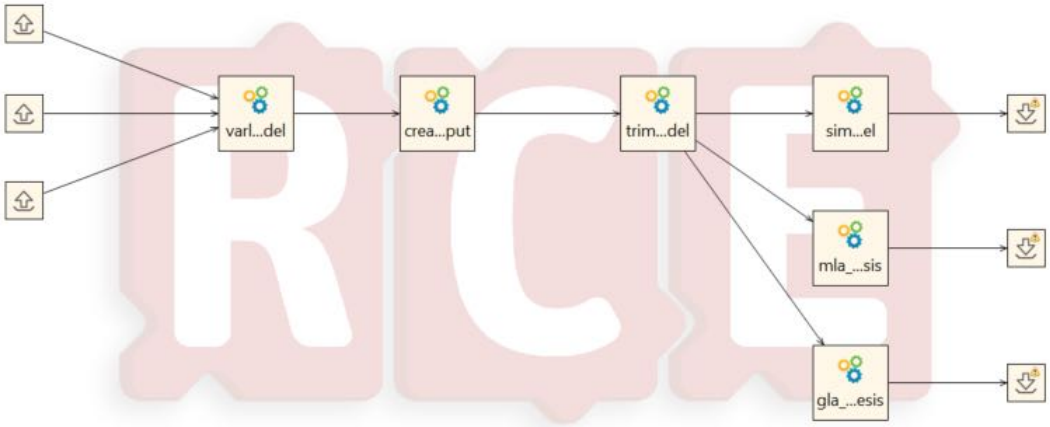


Figure 47: RCE workflow for the aeroelastic model generation and simulation

In the block ”trim lin model” the simulation environment is initialised and also linearized. It is possible to adapt the simulation environment based on various parameters, that have to be defined. First of all the model order is selected by deciding on a model with unsteady aerodynamics or steady aerodynamics, flexible dynamics or rigid dynamics. Furthermore, dynamics coming from sensors, actuators, airbrakes and the engine can be switch on or off. Dependent on the simulation to be performed or the type of controller to be synthesized gust inputs and load outputs can be added. Finally the operating point for which the aircraft model should be trimmed and linearized has to be selected by defining the indicated airspeed, the barometric height, the roll angle and others. Subsequent to the block ”trim lin model” the

model can be simulated with the block ”sim model” by means of the trim results. It creates a time series for dedicated inputs commanded to the control surfaces, the engine rotational speed and so on.

Models to be generated As already stated before, several flavors of the same model can be generated. The goal is to keep the number of the models as small as possible while satisfying the requirements of the control design blocks. The generated models are the following:

- Model for baseline control design – 12 state rigid model, set of LTI models parameterized by velocity;
- Model for manoeuvre load alleviation – LTI models;
- Model for gust load alleviation – LTI models;
- Model for flutter suppression controller design – low order flexible model, set of LTI models;
- Model for mission analysis – LTI and nonlinear model;
- Model for HIL tests and pilot training – real time capable nonlinear Simulink model.

Control Design Blocks In accordance with the workflow in Figure 47, after the ”trim lin model” block has finished, the synthesis of the various controllers follows. The linearised state-space systems offer the opportunity to synthesize linear controllers or gain-scheduled.

Baseline and Flutter Suppression Control Design Blocks The model generation, the control synthesis and the analysis of the resulting controllers for the baseline and flutter controllers is shown in the workflow presented in Figure 48.

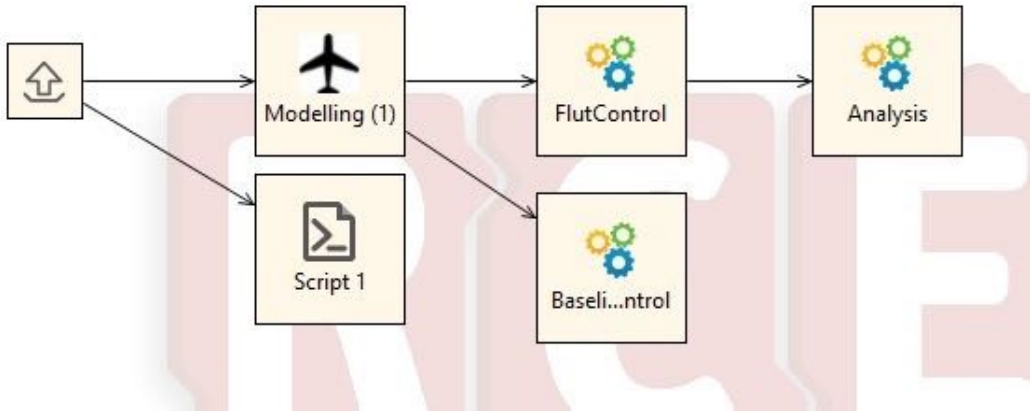


Figure 48: RCE workflow for the aeroelastic model generation for baseline and flutter suppression control design

The RCE workflow takes the nonlinear Simulink model with the configuring struct file from DLR-SR as the input files. These files are shared through RCE compressed files and are referenced withtin the CPACS file. The ”Modeling” block generates two models in this case. The flutter control synthesis block requires a low order aeroelastic model as an input. The aeroelastic model is obtained by the bottom-up

modeling approach ([107, 109]) which provides a sufficiently low order model for the control design. The generated model is a set if linearized model obtained from trimming and linearizing the simulink model. The linearized state space models are parameterized by the airspeed, but also by uncertainties in the FEM model of the aircraft. This model is saved as a mat file with name FlexACModel. The baseline controller accepts the rigid body, 12 state linearized models as input. This model is also generated in the ”Modeling” block from the aeroelastic moodel by residualizing the flexible and aerodynamic states. The baseline control design model does not contain any uncertainties, but a set of LTI state space model parameterized by the airspeed. This model is saved as a mat file with name RigACModel.

Based on the FlexACModel the flutter suppression controller is generated in the ”FlutControl” blocks ([86]) and sets the controller as a state space model at the output in a mat file. Based on the RigAC-Model the baseline controller is synthesized in the ”BaselineControl” block ([63]) and sets the simulink block with the configured PID controllers as the output.

Once the flutter suppression and baseline control design blocks finish the synthesis they pass the resulting controllers and the FlexACModel model to the ”Analysis” block. This block then runs a frequency domain analysis in two aspects. First, it assesses the performance of the two controllers acting together simultaneously. Second, it checks the robustness margins and flutter margins of the resulting controllers and if the minimum requirements are satisfied a pass flag is set and a PDF report is automatically generated. The pass flags and the PDF are finally set as the outputs of the block.

The main algorithms of each block and their adaptation to the MDO/RCE framework is given in deliverable D2.2 Report on tool adaptation for collaborative design.

MLA and GLA Control Design Blocks The second part of Figure 47 shows the manoeuvre load alleviation controller block ”mla control synthesis” and a gust load alleviation controller block ”gla control synthesis”. Both seek to reduce the wing root bending moment corresponding to manoeuvres and gust encounter. Their structure is predefined with specified inputs and outputs. The pitch angle and rate, the commanded and real vertical acceleration are needed for the manoeuvre load alleviation controller. Based on these measurements it calculates the necessary aileron and elevator deflections. The gust load alleviation controller takes the pitch rate, the vertical acceleration in the fuselage and on both wing tips as an input. It likewise provides aileron and elevator deflections. Both controllers are synthesized based on the structured H_∞ synthesis method with a full order model including unsteady aerodynamics, gust inputs and load outputs. Before the synthesis takes place, the order of the state-space model of the aircraft is reduced removing irrelevant dynamics. As an objective function for the MLA and GLA controller the weighted transfer function from gust input to wing root bending moment has to be reduced. Output of the RCE blocks are state-space models of the controllers. More controller types, like an active flutter suppression controller, could be synthesized subsequent to the ”trim lin model” block as well. The resulting controller state-space systems can then be fed to a closed loop model in order to analyse the overall aircraft performance.

Mission Analysis The frequency based analysis of the resulting controllers are carried out within the control design blocks. Therefore, the mission analysis can only be started in case all controllers have satisfactory performance and robustness. The mission analysis block takes the controllers and models as the input. The controllers are provided in a Matlab struc file from the control design block, the nonlinear model is given as a Simulink file with the configuration struc file. All the files are handed over via RCE as a compressed folder and are indexed in the CPACS file.

One of the goal of the mission analysis is to minimize the aerodynamic drag. Specifically, the induced drag is addressed by high aspect ratio wing designs. For this the induced drag has to be modeled for the reference aircraft and an optimal wing shape needs to be determined which results in the minimal induced drag.

The second goal is to assess the benefits of the improved aircraft and of the developed controllers. This goal requires a model that is of high fidelity, contains gust inputs and load outputs as well as all four controllers.

To analyse up to which speed it is safe to operate the aircraft, it is necessary to assess the speed at which flutter becomes unstable. For a flutter analysis the nonlinear aircraft model is linearized at several speeds. The poles of the model linearized at the highest speed are analysed at first. As flutter is most likely to be unstable at that speed, the analysis of the unstable poles reveal the flutter poles. Thus the flutter eigenvalues and eigenvectors are determined. By using a mode matching algorithm the flutter mechanism can be tracked for the linearized models with stepwise decreasing speed. With larger speed steps the tracking algorithm is more likely to fail. It is therefore recommended to choose small speed steps. The flutter speed is the speed, at which the flutter poles become stable. For more accuracy the flutter speed can also be estimated by interpolation between the flutter poles at the different flight speeds.

For the overall aircraft performance the aircraft is considered to operate in cruise. The flight conditions within cruise only changes due to defueling. To account for this change in mass a few discrete mass cases of the current aircraft configuration are provided. They represent different fuel levels. The necessary thrust for a mass case is then estimated by means of the overall drag, which is minimised through allocation of the control surfaces. The fuel consumption can be determined based on the required thrust. As soon as a certain level of fuel is consumed, a new mass case representing the predominant fuel level best is chosen. The sum of the distances the aircraft flies per mass case then provides the overall aircraft range.

Hardware-in-the loop Tests The second major toolchain in FLiPASED is the HIL test. The main purpose of the HIL test is to test the controllers running on the FCC - flight control computer. With this simulation, the FCC hardware and the controllers are testable and it can be assessed whether the designed controllers have any implementation limitation and also how they work in a realistic environment. The HIL architecture consists of two main components: a PC that runs the simulation model and the FCC running the control algorithm.

Requirements for the nonlinear model:

- The model is in continuous time and it has to run in real time. The real time capability of the model can be tested by running the simulation in External mode with Simulink Desktop Real-Time option set under menu item Simulation/Model Configuration Parameters/- Code Generation.
- The inputs of the model are the 19 controlled inputs. In addition to these inputs, the GLA controller tests also require the gust inputs.
- The outputs of the model are the sensors that can be used by the controllers. In addition to these outputs, the model also need to contain the loads as output in order to assess the MLA and GLA controllers.

Requirements for the Controller:

- The controller needs to be transformed to discrete time in case it was designed in continuous time. The sampling time is 5ms.
- The controller block designed by each partner has to be a static map between (U_C, $x_c[k]$) and (Y_C , $x_c[k+1]$), where x denotes the state of the controller (see Figure 49.). The input and output signals equal to the output and input of the model, i.e. $Y_C = U_M$ and $U_C = Y_M$.

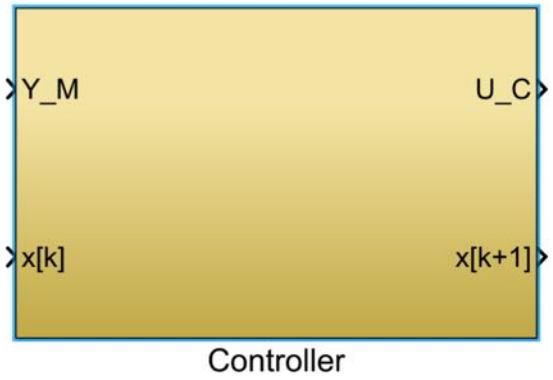


Figure 49: Controller inputs and outputs for the HIL test

The MDO toolchain needs to provide the model and controllers to the HIL environment. The model is provides as a Simulink model with fixed structure and a configuration file that contains all the necessary data of the model. The order of the HIL model is not reduced in order to retain as high fidelity as possible.

The control design blocks of the MDO toolchain provide the baseline, MLA, GLA and flutter suppression controllers in discrete time. All the controllers are given as state space models that are ready to be used for automatic C code generation in order to be uploaded to the FCC.

The HIL tests provide time domain simulation results after evaluating the controllers. The responses are evaluated numerically and based on the evaluation pass/no pass flags are set for each individual controller.

Flight Tests The third major toolchain consists of pilot training, ground tests and flight tests. The controllers and the models are the passed on from the HIL tests which can directly be used for pilot training. Ground tests and flight tests can only be carried out with the physical aircraft and not after each iteration of the MDO toolchain.

The ground testing serves to evaluate the structural properties of the newly developed wing in order to clear its airworthiness and to produce a comprehensive modal model of the aircraft. This modal model can then be used for Finite Element (FE) model updating,flutter calculations and controller updating.

The main goal of the flight test is to validate the maturity of the developed controllers and control design methodologies. The baseline controller has already been validated in the legacy FLEXOP project. Therefore, the main focus in FLiPASED is on the testing of the MLA, GLA and flutter suppression controllers.

The details of the ground and flight test plans are given in deliverable “D1.3 Demonstrator Ground and Flight Test Requirements Definition”.

Overall Architecture Evaluation The goal of this section is to explain the connection between the MDO, HIL test and flight test toolchains.

The MDO toolchain is the main block in this case which has its own optimization and gets back to the CPACS generation block after each iteration. In this tool one of the main focus for the model generation, model reduction blocks and control design blocks are the robustness aspects of the underlying algorithm. These need to run automatically, without human interaction in the presence of changes in the

aircraft. This comes at an expense that the individual controllers do not achieve the highest possible performance. The other main goal of the MDO toolchain is to show the improvements of the optimization, which involves aircraft geometry, sizing, modeling and control design simultaneously, with respect to the reference aircraft. The HIL test and flight test block come as an auxiliary tool to validate the developed methodologies.

The HIL tests evaluate the implementation aspects of the controllers and serve as a final step before flight testing the controllers.

The flight test goals are twofold in case of the MDO toolchain. The main goal is to validate the control design technology maturity. This is especially valid for the MLA, GLA and flutter suppression controllers of the project since they not have been flight tested yet (using the model based design methodology within FLiPASED). The other goal of the flight tests, as opposed to the MDO toolchain, is that the resulting controllers can be fine tuned by ”hand” to achieve optimal performance and provide lessons-learnt to the designers and to the aviation community in general. In this case the robustness of the synthesis algorithms to be able to be run in an automatic manner is not of a paramount criterion. In addition, the fine tuning of the controllers is to be done based on the aircraft model that has been updated via flight test data.

At the end of the cycle, the lessons learned from the HIL tests and flight tests need to be fed back to the MDO toolchain. This is done via engineering considerations. If the HIL tests indicate that some controller has implementation difficulties, the corresponding control design algorithm needs to be updated. Similarly, if the flight tests show that a controller has lack of performance or robustness during flight tests, the algorithms need to be adjusted as well.

The main output of the deliverable is the definition of the functional division, data flow and specific data types exchanged among the partners in the collaborative design. This is especially important in case of the types of models generated throughout the workflow since one of the key goals of the project is to reduce the overall number of models in the development. The other main result is to connect the ”lessons learned” from HIL and flight test to the MDO toolchain to be able to update the underlying algorithms if required.

The MDO tools are being integrated into the RCE framework by the respective tool owners based on the interface definitions laid out in the deliverable. Once the integration is finished the MDO toolchain will be tested and fine-tuned by the consortium. The present workflow is developed for the demonstrator, but the overall methodology is almost the same for the scale-up task within WP4. The main difference comes from the objective function and the inner convergence loop for structural sizing - aeroelastic tailoring, what is not present in the demonstrator workflow, where only a structural check is established.

Reference Model Definition The deliverable “D1.5 Reference Model Definition” lays the foundation for the scale-up task in WP4 of the project. In the beginning of the project, several key factors have been identified and objectives as well as performance metrics have been proposed to show the benefits of the MDO tool-chain developed within the project. The insights gained in the FLiPASED project during the flight test and the experience with the method and tools used for the design of active control technologies will then be applied to the design optimization of a full-scale aircraft. This document explorers the reference model alternatives, which are available for the research teams within the project. The model has to be suitable to apply the active control technologies and representative enough to show the benefits of the envisaged aero-servo-elastic optimization framework. During the optimization, a derivative aircraft based on the reference model will be designed. The pros and cons of the individual models will be detailed and the rationale for the final model selection will be presented.

Motivation In order to show the benefits of including the Flight Phase Adaptive Aero-Servo-Elastic Aircraft Design Methods (FliPASED) in an integrated aircraft design, it is planned to demonstrate the performance claims in a scale-up task. As baseline reference for this scale-up task a Flexible Aircraft Benchmark will be defined in coordination with the industrial advisory board and used as the reference during the project. The resulting derivative aircraft will have a higher aspect ratio and therefore a more flexible wing. Aeroelastic tailoring will be applied to the CFRP wing structure in conjunction with active control augmentation, which is enabled by advanced avionics and a flight control architecture. Advanced Manoeuvre and Gust Load Alleviation functions will allow for a significantly reduced wing structural weight. Since high aspect ratio wings are more prone to flutter instabilities within the certification envelope, an Active Flutter Suppression will allow for further weight savings compared to classical open loop designs. Wing shape control reduces the drag in off design flight conditions and further increases the efficiency. The two main objectives of the scale up task are the demonstration of the applicability of the collaborative design process to a (full-scale) passenger aircraft and the quantification of the benefits of integrated aircraft and controls design in terms of structural weight reduction and aircraft over-all performance parameters. A comparison of traditional aircraft conceptual design can be seen on Figure 50, where aerodynamics and structures are optimized separately in a sequential order, and the resulting design will be sub-optimal (as shown in Fig. 51). It is well known now that coupled aero-elastic design should be done in a MDO framework, however very few results are available on coupled aero-servo-elastic MDO process, which is the key goal of FLiPASED.

Scope of Scale up Task The focus of the FliPASED project is on including control design as a primary discipline in a collaborative design workflow. Some previous experience is available within DLR, where a comprehensive load analysis process [55] is already included in projects like Digital-X and Victoria [31, 32]. Also, preliminary steps have been taken to consider active control systems within the design cycle [38]. The efforts within the FliPASED project mainly target the inclusion of the control technologies in the design workflow, while deemphasizing the aerodynamic design. The aerodynamics will consist mainly of low fidelity aerodynamics and methods based on potential flow theory. Hence, transonic effects like shocks and wave drag will not be considered in the scale-up task. This is a conscious decision in order to avoid overlap with other projects and to allow quick calculation times. Furthermore, no emphasis is placed on the choice of a particular MDO architecture. This distinguishes the approach in FliPASED compared to other efforts which mainly focus on aero-structural optimization [46] and therefore will demonstrate complementary capabilities. In the future, the findings of FliPASED may be integrated in MDO workflows, where more realistic aerodynamic properties are considered. In the project FliPASED the benefits of including active control technologies early in the design will be demonstrate rather than considering them as an afterthought.

Scale Up Objective Function The overall objective function for the scale up task will be based on evaluation of mission criteria, such as range or blockfuel. This way two primary design goals can be addressed. The first goal is to minimize the aerodynamic drag. Specifically, the induced drag is addressed by high aspect ratio wing designs. However, the resulting slender wing structures tend to be very flexible and defueling the wing tanks change the mass distribution and in turn the shape of the wing. To counteract the detrimental effect on the induced aerodynamic drag, active wing shape control deflects the control surfaces to restore a drag optimal lift distribution for the changing wing mass. The second goal is to minimize the structural weight. This can be achieved by employing active load alleviation control laws to minimize design loads for manoeuvres as well as gusts and turbulence in combination with passive methods for load alleviation such as aeroelastic tailoring. Furthermore, the aforementioned high aspect ratio wings are more prone to an adverse fluid structure interaction called flutter. Conventionally, this is addressed by increasing the wing stiffness or placing additional mass in suitable locations. The employment of active flutter suppression allows to relax these stiffness requirements and therefore save weight. To assess the benefits of the mentioned active control technologies,

the mission is analyzed at multiple points of the flight envelope and via various mission profiles, i.e. different mass cases due to defueling. The conjecture is that inclusion of active control theory in the design phase leads to very different wing designs and a large overall fuel savings.

Differences between Demonstrator and the Scale up Workflow The workflow that is setup in WP2, initially addresses the design of wings for the demonstrator. The objective there is to maximize the difference between open loop and closed loop performance of the individual control functions in order to assess and validate their benefits by flight test. Fuel burn and minimal weight are not primary design objectives. For the scale up task, a passenger aircraft is considered. The design objectives have been described in the previous section. Apart from the differing objective functions, the most notable difference of the demonstrator workflow, is that the structure is now sized by the loads, i.e. the employed control functions have a direct impact on the overall weight of the structure. The updated stiffness and mass properties therefore make a convergence loop necessary. Figure 52 shows an early version of the envisaged scale-up workflow. The XDSM diagram shows a convergence loop including structural sizing, controller design of the various functions and the loads analysis of the closed loop aircraft.

A further complication arises, as the CATIA based structural model generation is targeted towards the demonstrator wing. It will be investigated how this model generation process can be adapted to a transport aircraft wing. As contingency, an alternative model generation module (CPACS-MONA) is available at DLR Institute of Aeroelasticity. This module has been used in several MDO workflows before.

Reference Model Candidates The intention of the scale-up task is to start from an already feasible, optimized aircraft baseline and show the potential benefits of the ASE MDO workflow with respect to the current state of art. For this reason it is important to have a model which includes all the components necessary for aerodynamic, structural and control evaluation. The team do not want to design a new aircraft, just apply control design technologies to a high aspect ratio variant of the reference model. For the scale-up task, the following models were considered as potential reference configurations. A brief description of each of the models together with their potential benefits and drawbacks are listed below.

XRF1: Airbus eXternal Research Forum Model (A330 like) The XRF1 Model is a multidisciplinary aircraft model which is intended to further development and validation of flight physics and broader multi-disciplinary technologies by the external research community. The XRF1 model can be released to research establishments under the terms and conditions of a Framework Non Disclosure Agreement (FNDA). The DLR used this model in several MDO related projects and the FP7 EU project Smart Fixed Wing Aircraft. A parameterization in CPACS format is available and could be used.

For:

- Experience at many research establishments across several projects with the XRF1 model
- Mature aircraft dynamic model
- Has also been used for scale-up studies in FLEXOP

Against:

- NDA required from partners using the model
- Rules pertaining to IT security apply
- Restrictions on publications apply

CRM: NASA Common Research Model (B777 like) In order to improve the state-of-the-art in computational fluid dynamics, Langley Research Center and Ames Research Center of NASA joined forces to produce data sets using the same research model – the Common Research Model. Using the same Mach numbers and model configurations, they have been able to gather data that is provided to the worldwide research community. One of the main aim of the CRM model is to investigate CFD methods, hence the Common Research Model Wing/Body and Wing/Body/Tail configurations have been used on the drag prediction workshops of NASA since 2009. Details of the model are initially reported in [115], but further research expanded the model to a higher aspect ratio version (uCRM-13.5) for very flexible wing design studies. The following components are available as open source:

- Geometry files for the wing-body-htail configuration of each aircraft (IGES/TIN)
- Aerodynamic mesh files for the wing-body-htail configuration of each aircraft, both in multi-block and overset format (CGNS)
- Structural mesh files for the aluminum wingbox structure including material properties based on a smeared stiffness blade-stiffened panel approach, external control surface and engine masses, and aerodynamic loads for nominal cruise (BDF)
- Reference solutions using the MACH framework and NASTRAN

For:

- Free-to-use CAD model of aircraft
- Structural model available at DLR-AE (FERMAT configuration)
- Aero-loft suitable for high-fidelity CFD

Against:

- CPACS dataset unavailable
- Lesser experience with this configuration in the consortium compared with the other models
- Boeing/NASA-initiated model

D150: DLR 150Pax Model (A320 like) The D150 configuration was developed within the DLR project VAMP [127]. It is comparable to the Airbus A320-200. Data published by the manufacturer, for example on the Airbus website, and input data to the preliminary design program PrADO for the application example Airbus A320, are used for the D150 configuration [54]. Its geometry is shown in Figure 55.

Table 3 lists the general parameters of the D150 configuration. The cruise speed V_C and cruise Mach number M_C are set to the maximum operational speeds V_{MO} and M_{MO}. The values for V_{MO} and M_{MO} for the Airbus A320 can be found in the EASA Type-Certificate Data Sheet [27]. The dive speed V_D can be calculated using the diagram of worksheet LTH BM 32 100-05 of the Luftfahrttechnischen Handbuch (LTH), and the dive Mach number $M_D = M_C + 0.07$ from the Acceptable Means of Compliance AMC 25.335(b)(2) of CS25.

The three airfoil profiles used for the four profile sections, using which the planform geometry is built, originate from the geometry of the DLR-F6 configuration. The DLR-F6 configuration is similar to the geometry of the Airbus A320 and was developed in the 1980s as a publicly-available geometry for aerodynamic studies.

For:

Wing	
Surface area	$122.3 m^2$
Span	$33.91 m$
Reference chord	$4.19 m$
Aspect ratio	9.4
Taper ratio	0.246
Sweep angle at 25% chord line	$24.94°$
HTP	
Area	$30.98 m^2$
Span	$12.45 m$
Aspect ratio	5.0
Taper ratio	0.33
Sweep angle at 25% chord line	$28.0°$
VTP	
Area	$21.51 m^2$
Span	$5.87 m$
Aspect ratio	1.6
Taper ratio	0.35
Sweep angle at 25% chord line	$35.0°$
Operational empty weight (OEM)	$40638 kg$
Maximum zero-fuel weight (MZFM)	$60500 kg$
Maximum take-off weight (MTOM)	$72500 kg$
Cruise Mach number	0.78
Cruise speed / Mach number	$180 m/s$ EAS, Mach 0.82
Dive speed / Mach number	$209 m/s$ EAS, Mach 0.89
Maximum flight level	$12500 m$

Table 3: Main parameters of the D150-configuration

- DLR-proprietary configuration
- Relevance to industry - short/medium-range (SMR) configuration
- CPACS dataset available and maintained across various project developments
- Experience from several other projects involving D150 model
- No restrictions pertaining to publication

Against:

- Aero-loft not suitable for CFD simulations - aerodynamics restricted to potential flow methods

Reference Model Choice and Impact From the considered choice of models discussed in the earlier chapter, the DLR-D150 was selected by the consortium as the preferred reference model for the scale-up task.

The primary rationale for choosing the D150 is its relevance to industry and parallel on-going research activities in different projects, ie. in a next-generation SMR aircraft. An A320-like configuration is considered to be short and medium range and well-suited for this classification. Moreover, the D150 being a DLR-proprietary model, the availability of a CPACS dataset and freedom pertaining to publications are advantageous.

The drawback of not having a good enough aero loft to carry out CFD simulations as in the case of the D150, is mitigated by the fact that only potential flow methods are intended to be employed. The target performance optimization goal in FLiPASED is only the reduction of induced drag, i.e. drag due to lift distribution and not wave drag and airfoil optimization.

Relevance to research community/industry The decision to choose the DLR-D150 is in line with multiple local on-going initiatives and projects. Among others, one can count:

- VirEnfREI-DLR - LuFo funded project involving DLR and Airbus. The project involves establishing an MDO framework for aircraft design, considering industrial requirements and its application to the design of an SMR aircraft. The optimized configuration is to be tested under flight conditions in a transonic wind-tunnel.
- MuStHaF-DLR - LuFo funded project involving DLR institutes. The project is targeted towards future high aspect ratio SMR aircraft configurations considering different wing technologies - multi-functional control surfaces, control algorithms for active flutter suppression, online flutter stability monitoring, among others. A selection of the developed technologies are to be tested in a flying demonstrator of a scaled SMR aircraft wing.
- MAJESTIC - DGAC funded project involving ONERA and Airbus. It is concerned with the aeroelastic modelling methodology and control design for flutter phenomena. The considered use-case is a generic single aisle high aspect ratio configuration.

Apart from this, Dassault-Aviation, a member of the Scientific Advisory Group in FLIPASED, had expressed interest during the initial phase of the project in a potential narrow-body aircraft for scale-up studies as opposed to wide-body aircraft, given their product portfolio in business-jets.

Impact of reference aircraft on other WPs The consortium also considered the impact of reference model choice on the rest of Tasks in ever WP - at least in a broad manner depending the choice of Single Aisle or a Wide Body aircraft was discussed and an unanimous decision was made to focus on a single aisle aircraft what is more relevant to the industrial partners of the consortium members.
WP1 is unaffected by the choice of the reference aircraft. Task 1.3 Collaborative Work Process is the same for the conceptual design of a business jet, single or twin aisle aircraft. The MDO work process has slightly different mission profile but that is only a small parameter change in the overall framework. Task 1.4 Data Analytics for Model Validation is also unaffected by the choice of aircraft, since it only focuses on analyzing the results. It might be possible that the consortium is able to achieve lower fuel efficiency improvements due to shorter wingspan or lower number of individual flight control surfaces fitted to the wing (in comparison to a widebody airplane), but the analysis tools will be unaffected. Within WP2 several tasks are connected to both the demonstartor and to the scale-up task, namely Task 2.1: Tool Adaptation: Structural Design, Task 2.2: Tool Adaptation: Aerodynamics, Task 2.3: Tool Adaptation: Aeroelasticity, Task 2.4: Tool Adaptation: Movables Design, Task 2.5: Tool Adaptation: Control Design . These are all using the same software framework for the demonstrator and the scale-up workflow, but their parameters and their fine tuning are different. These generic tools have for example the aircraft geometry (CAD) as an input parameter and they provide outputs based on the user defined tuning knob settings. For example the FEM model might have condensation points every 10 cm or at every 100 cm. Hence a large 65 m wingspan aircraft might be represented by fewer condensation points than a 7 m wingspan demonstrator. Also, the number and location of the sensors and flight control actuators are just a parameter for the on-board, model-based, flight control system. The tools developed within WP2 are generic in a sense that both workflows (and different aircraft configurations within each workflow) use them with the adequate parameter settings. It might be possible that in the demonstrator workflow fuel level and c.g. position do not play such an important role, that every model and every tool has to account for fuel variation, but changes in the velocity are already captured and hence the tools are meant to handle parameter variations within the workflows. Within the scale-up workflow these variations are more pronounced but they are only quantitatively different no fundamental change are foreseen between them.

WP3 contains all activities related to the physical testing of the demonstrator. The overall activities are performed to validate the predictions and provide feedback about the performance of the tools within the MDO toolchain. There is no direct feedback between the demonstrator flight test results and the scale-up task. It is the aim of the consortium to mature the tools via lessons learnt within the flight test campaign, as seen in Figure 56, but it is not possible to characterize the type and impact of the feedback before evaluating the toolchain results and the demonstrator flights. The impact on the scale-up workflow is even more distant, since lessons learnt during the flight test will provide indirect feedback to a large SMR or widebody aircraft, hence the choice of reference aircraft being 70 m or 35 m in wingspan has no direct impact on the tasks within WP3.

Tasks within WP4 are directly impacted by the choice of the scale-up model, and since the project is delayed due to difficulties in the flight test campaign, as well as due to the pandemic, the consortium selected the model which involves the least amount of uncertainty. This being the DLR internal D150 model, where Task 4.1: Aircraft design objectives is significantly helped by the ongoing and newly launched projects of DLR and ONERA, where the interest of their industrial partner Airbus lies in the SMR aircraft domain. It is foreseen that synergies between FLiPASED and these projects could be leveraged and design objective setup will receive feedback from Airbus and Dassault. Task 4.2: Implementation of reference A/C data into tool chain is also heavily impacted by the choice of this decision, since large part of the D150 dataset are already in the CPACS format, what is the descriptor language for the FLiPASED toolchain. Moreover, both DLR-AE and DLR-SR has working experience with these models. In principle the most profound changes in the existing D150 and the one needed for the demonstration of enhancements in FLiPASED are the addition of flaps, sensors and actuators on the wing.

These have to be incorporated into any scale-up aircraft model, since public models of the XRF1 and CRM both have the standard, limited number of, flight control surfaces and no inertial sensors within the wing. These additions will be incorporated into the D150 derivative, where minimum size of actuators and wing thickness might restrict the consortium to split the most outer ailerons into 2 pieces instead of 4 individual pieces, what could have been feasible on an A350 sized wing. The consortium is well aware of the fact that even 8 individual trailing edge primary flight control surfaces on an SMR aircraft will lead to more optimized wing shape, and will allow more tailored load alleviation, as well as flutter mitigation and drag optimization in comparison to the single aileron on the A320 wing (see Fig. 57). While it might be possible to fit 16 ailerons to the trailing edge of an A350 size aircraft (see Fig. 58), the incremental effects of 8 vs. 16 ailerons on the wing will be less pronounced than fitting 2 vs. 4 ailerons [12].

Task 4.3: Development of avionics Systems Architecture for reference A/C will be also mainly unaffected by the choice of the reference aircraft. As stated above the size, weight and power requirements of the actuators fitted in a lower thickness SMR aircraft might allow less individual control surfaces (i.e. 8 instead of 16) but we foresee a highly over-actuated system with large number of redundant control surfaces where similar issues have to be solved in the 8 or 16 actuator case. On the other hand we do not see a similar limiting constraint in the sensor placement problem. Task 4.4 concerns the design study itself. Since SMR aircraft has lower range it might be beneficial from simulation time perspective to choose this instead of a long range aircraft. It is not clear for the consortium at the moment what type and how many simulation runs will be performed after each iteration cycle, but the overall methodology with distinctive load cases and gust encounters to assess the performance of the load alleviation functions will be the same irrespective of the aircraft type. We intend to run hundreds of simulation points instead of the few cases listed in the certification requirements of EASA, since the active control functions can be evaluated only in a dynamic setting. System benefit assessment (Task 4.5) will be also mostly unaffected by the choice of medium or long range aircraft, since the baseline performance and the outcome of the optimization, in terms of performance gains, increase in complexity, certification effort, and overall design effort will be compared.

Conclusion The Deliverable D1.5 pertains to the selection of a reference model for the scale-up task in WP4 within FLIPASED. The scale-up task involves an integrated aircraft design workflo, enabled using an MDO approach involving aeroelastic tailoring for the optimization of the wing structure in conjunction with active control augmentation for load alleviation, flutter suppression and wing shape control, leading to direct drag reduction.

The DLR-D150 model is chosen as the baseline reference for this scale-up task. The primary motivation behind the selection is its relevance to both industry and parallel on-going projects along several national fronts, ie. in an SMR aircraft, as well as its maturity and availability for the consortium members. The studies performed within the scale-up will be beneficial in demonstrating the benefits of including mature-levels of active control technologies right from an early preliminary design phase of aircraft development, rather than considering as a subsequent design step inherently leading to more sub-optimal solutions.

Data Analytics for Model Validation The following results are based on preliminary findings.

The deliverable ”D1.6 Data Analytics for Model validation” focuses on comparing results and findings coming from different sources. The main reason to have specific assessment of results coming from theoretical models or experimental tests is to build confidence in the developed tools and methods. The data from flight tests will serve as a baseline to validate structural dynamics, aerodynamics, controls and avionics instrumentation models. Analysis tools with standard validation routines will be provided in Nastran and Matlab environment for structural dynamics and controls respectively. These tools along

with Python based data science software will be used within the project and the underlying theory along with interfaces of these tools are documented in D1.6.

The project goal, set in the proposal within Task 1.4 Data Analytics for Model Validation (SZTAKI, DLR,ONERA,TUM) aims at: "*Significant part of the engineering effort is devoted in research projects to provide the adequate interface for tools and methods developed in prior projects. These tools and the corresponding analysis steps with their interfaces will be part of the open data initiative, to provide seamless access to the core problems with adequate analysis tools to the research community. The data from flight tests in the early part of the project, provided by re-using the demonstrator platform developed within FLEXOP, will serve as a baseline to validate structural dynamics, aerodynamics, controls and avionics instrumentation models. Analysis tools with standard validation routines will be provided in Nastran and Matlab environment for structural dynamics and controls respectively. During the second half of the project these tools will be expanded to address the attainment of MDO criteria, during the development cycle – including the functions of weight reduction, fuel efficiency and gust load alleviation. Based on the large amount of simulation and experimental data both analytical and data driven approaches will be pursued for model predictive control, function shape fitting by support vector machines and deep learning, parameter search by Monte Carlo methods, and more. The project will also use Python based data science software, including numpy, scipy, pandas, scikit-learn, Tensorflow, Keras, matplotlib and many more, in Jupyter notebooks, as the emerging de facto standard sharing and collaboration tool for data scientists.*"

Overall Architecture and Tools to connect MDO and Testing This section describes the overall structure of MDO toolchain and the tools used there. A short introduction regarding each blocks is also given. The connection between the MDO, HIL test and flight test toolchains is also explained in this section.

Overall Architecture and Tools of MDO Toolchain The MDO toolchain is the main block in this case which has its own optimization and gets back to the CPACS generation block after each iteration. The main goal of the MDO toolchain is to show the improvements of the optimization, which involves aircraft geometry, sizing, modeling and control design simultaneously, with respect to the reference aircraft. Figure 130 shows the overall architecture of MDO toolchain.

The following sections will give a brief introduction of function blocks in MDO toolchain and the used standard tools. For more informations please refer to previous deliverables 1.2, 1.4, 2.2 and 4.1.

CPACS The data model CPACS has been introduced and developed at the German Aerospace Center (DLR) since 2005. CPACS is implemented in XML. Making use of the hierarchical representation of data in XML the structure of CPACS mainly follows a top-down approach which decomposes a generic concept (e.g., an aircraft) into a more detailed description of its components. This originates from the conceptual and preliminary design of aircraft, where the level of detail is initially low and continues to increase as the design process progresses. The hierarchical structure furthermore promotes the simplicity of the exchange format which is required in collaborative design environments so that the various stakeholders can easily append their results. CPACS serves as the data model in this toolchain.

RCE DLR's Remote Component Environment (RCE) [11] is an open-source software environment for defining and executing workflows containing distributed simulation tools by integrating them into a peer-to-peer network. In this toolchain, RCE is used as the integration platform.

CPACS generation block CPACS generation block, which is the first block in the MDO toolchain, aims to generate the first version of CPACS file with a Python script.

Geometry block Geometry block aims to update the Catia model based on the incoming CPACS file from the upstream CPACS generation block.

FE-model block The function of the FE-model block is meshing the geometry model and assigning structural properties. A Splining model, which couples the structural and aerodynamic model, is also generated in this block.

Aero-model block The aero-model block takes the geometry definition in CPACS file as input, generates the DLM aerodynamic model, and exports it to a Nastran bdf file.

Aeroelastic Model Generation and Simulation Based on the aerodynamic, structural and spline grid information as well as mass and stiffness matrices, simulink model will be generated in this block and will be used for control synthesis design.

Baseline and Flutter Suppression Control Design Blocks This block takes the nonlinear Simulink model with the configuring struct file from DLR-SR as the input files and generates two models in this case, one for the flutter control synthesis block and one for the baseline controller. Based on the model the flutter suppression controller and the baseline controller are generated. Once the flutter suppression and baseline control design blocks finish the synthesis, a frequency domain analysis will be ran to assesses the performance of the two controllers acting together simultaneously, to check the robustness margins and flutter margins of the resulting controllers.

In this MDO toolchain one of the main focus for the model generation, model reduction blocks and control design blocks are the robustness aspects of the underlying algorithm. These need to run automatically, without human interaction in the presence of changes in the aircraft. This comes at an expense that the individual controllers do not achieve the highest possible performance. This is also where testing comes into play to validate the developed methodologies.

Connection between MDO Toolchain and Testing The HIL test and flight test blocks serve as auxiliary tools to validate the developed methodologies, as shown in figure 172.

The HIL tests evaluate the implementation aspects of the controllers and serve as a final step before flight testing the controllers.

The flight test goals are twofold in case of the MDO toolchain. The main goal is to validate the control design technology maturity. This is especially valid for the MLA, GLA and flutter suppression controllers of the project since they have not been flight tested yet (using the model based design methodology within FLiPASED). The other goal of the flight tests, as opposed to the MDO toolchain, is that the resulting controllers can be fine tuned by ”hand” to achieve optimal performance and provide lessons-learnt to the designers and to the aviation community in general. In this case the robustness of the synthesis algorithms to be able to be run in an automatic manner is not of a paramount criterion. In addition, the fine tuning of the controllers is to be done based on the aircraft model that has been updated via flight test data.

At the end of the cycle, the lessons learnt from the HIL tests and flight tests need to be fed back to the MDO toolchain. This is done via engineering considerations. If the HIL tests indicate that some con-

troller has implementation difficulties, the corresponding control design algorithm needs to be updated. Similarly, if the flight tests show that a controller has lack of performance or robustness during flight tests, the algorithms need to be adjusted as well.

More details regarding toolchain validation will be given in the following sections.

Structural Dynamics Model Validation The tasks related to structural dynamics of the aircraft models are led by DLR-AE, but contributions are made by ONERA, TUM, SZTAKI and DLR-SR as well.

The main steps regarding the task are:

- structural model development and GVT based update
- Model comparison and fine tuning for RCE toolchain based and GVT based model matching
- Operational modal analysis based model update during flight tests and its connection how this feeds back to NASTRAN models
- Description of used tools and how they can we standardized

In this chapter, a summary of the structural dynamics model and the model-updating activities pertaining to its update are described.

NASTRAN structural dynamic model The structural dynamic models of the T-Flex aircraft are developed using a modelling toolchain established during FLEXOP and FLIPASED. In total, three pairs of wings are designed, manufactured and tested on the UAV test-bench:

(i) wings -0 - a pair of wings optimized using balanced-symmetric type of laminates serving as the reference wing

(ii) wings -1 - a pair of flutter wings designed to trigger flutter within the test-regime, whose flight envelope will then be extended using active flutter control

(iii) wings -2 - a pair of wings optimized using unbalanced composite laminates, to demonstrate passive load alleviation through aeroelastic tailoring

The structural FE models for the wing pairs -0 and -2 are generated using an in-house model generator ModGen at DLR-AE [53], while those of the -1 wing are obtained from a CAD-FEM toolset at TUM. The wing models are integrated to the fuselage and empennage models generated during FLEXOP at DLR-AE. The fuselage and empennage models are also generated using ModGen [53].

Model-updating of the -0 wings A ground-test campaign [101] involving structural tests and ground vibration tests (GVT) has been performed on the T-Flex aircraft. An update of the FE model of the -0 wing has been performed based only on experimental data from static tests, while an update using data from the GVT is being studied at present.

The static test was performed with the main objective being the assessment of the stiffness properties of the wing and validation of the pertinent structural models developed. Figure 61 shows the deflection of the wing-tip as a function of the applied tip-load. Shown in Figure 62 is the span-wise displacement of a wing-half subjected to 3kg load at the tip, comparing the static tests and the initial FE model. The observed difference in the displacement could be attributed to several factors including modelling assumptions and simplifications, manufacturing deviations, material scatter, etc.

Mode	GVT (Hz)	FE (Hz)	Δf (%)
2n_wing_bending-s	3.37	3.27	-2.9
3n_wing_bending-a	8.28	8.35	0.9
1n_wing_inplane-a	8.88	18.45	-
4n_wing_bending-s	12.12	11.86	-2.1
tail_rock-a	17.32	-	-
1n_wing_inplane-s	19.26	18.09	-6.1

Table 4: Comparison of eigen frequencies of the flexible modes: GVT vs stiffness-updated FE model of the -0 aircraft (*in* - *i* nodes in the mode, s - symmetric, a - antisymmetric)

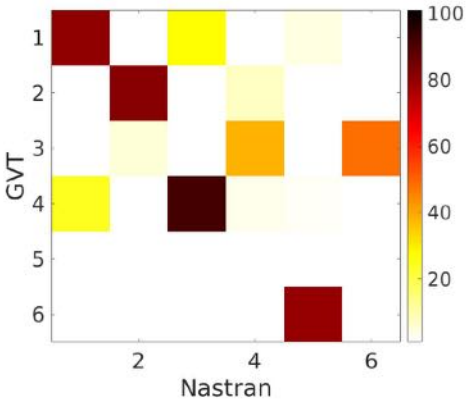


Figure 63: MAC matrix: GVT vs stiffness-updated FE model of the -0 aircraft

The present version of the model-updating is performed by introducing a knock-down on the engineering stiffness (E_1, E_2, G_{12}) in the FE model of the wing and in the clamp used for the wing attachment. An alternative approach through the inclusion of tuning-beams and optimizing its properties has also been attempted. A comparison of the frequencies between this stiffness-updated FE model and the GVT results is shown in Table 4. Also shown is the modal assurance criterion (MAC) which is an indicator of similarity between mode shapes from two sets in Figure 63. It is seen that the FE model captures the out-of-plane bending behaviour of the wing well. On the other hand, the in-plane behaviour of the wing and the stiffness and mass modelling of the fuselage and empennage need to be investigated in more detail.

The stiffness-updated structural model serves as the basis for generating a next iteration of ASE models for controller synthesis. In the next steps, a more refined approach at model-updating will need to be performed considering other possible sources of deviation such as an improved modelling of wing-fuselage joint, localized stiffness-updates and updated mass-modelling while utilizing also the frequencies and mode-shapes obtained from the GVT.

Comparison of -1 aircraft structural dynamic model with static test Static test of -1 wing was conducted in the FLEXOP project at the same time as -0 and -2 wing to verify the stiffness properties of manufactured wing and validate the FE-Model developed during design stage. Figure 173 shows wing tip deflection at different load cases and their linear fit. Linear stiffness property can be clearly seen in the figure 173. Because of measurement error, there is zero drift when the load was increased from zero and decreased to zero again.

FE-model is elaborated to replicate the static test. Figure 174 shows the comparison of span-wise displacement of wing under 5 kg tip load between simulation and test. The manufactured wing is more flexible than it modelled. It shows same trend as the -0 and -2 wing. The deviation between simulation and test is around 12%, without consideration of zero drift in the test.

Same investigation was made for the torsional load cases. Figure 175 shows the linearity of the model under the torque loads. Figure 176 shows the comparison of span-wise torsion of wing under 2 kg torque load between simulation and test. There are only 0.1 deg differences. Taking the measurement error into account, the results match quite well.

Mode	GVT (hz)	FE (hz)	Δf (%)
2n_wing_bending-s	2.94	2.91	-1.02
1n_wing_in-plane-a	7.01	–	–
3n_wing_bending-a	7.57	8.15	7.66
wing_torsion-s	10.27	10.50	2.24
wing_torsion-a	10.73	10.61	-1.12
4n_wing_bending-s	12.13	12.11	-0.16
2n_wing_in-plane-s	15.07	15.06	-0.07

Table 5: Comparison of eigen frequencies of the flexible modes: GVT vs FE model of the -1 aircraft (*i*n - *i* nodes in the mode, s - symmetric, a - antisymmetric)

Comparison of -1 aircraft structural dynamic model with GVTs The -1 wing FE model is generated using a CATIA - Hypermesh toolset at TUM. The model is of a very high fidelity comprising of detailed elements for the structural as well as non-structural entities such as on-board systems.

A comparison of the eigen frequencies of the -1 aircraft model (without update) and the GVT is shown in Table 18. It is seen that a generally good agreement between the FE model and the GVT results exists. Two observations can be made with respect to this comparison.

The third flexible mode (3n_wing_bending-a) has the largest different in the experimental results with respect to the GVT. Given that this wing bending mode participates in the critical flutter mechanisms of the -1 aircraft, it is important to update the wing FE model with respect to the frequency of this concerned mode. Secondly, the second flexible mode (named 1n_wing_in-plane-a) which is observed during the GVT but not in the FE simulations appears to involve a relative motion between the fuselage and wing as shown in Figure 177. Such a mode is expected due to some free-play or softness in the attachment between the fuselage and wings, which is not tuned for in the FE models where an idealized attachment is assumed. In order to be able to simulate this mode, one approach would be to tune soft springs at the wing-fuselage interface such that the mode is present in the simulation. Both the FE model update mentioned in the former and a study on how to introduce the concerned mode discussed in the latter are being studied at present. An approach using so-called tuning beams is planned for this task.

Model-updating of the -1 wing The model updating of -1 wing is first conducted with static test data. Knock-down factor is applied on the engineering stiffness (E_1, E_2, G_{12}) of the wing skin and spar. The model updating is based on the 3 kg bending load case. As you can see from Figure 178, the simulation result matches quite well with test data. The deviation with the test result is reduced to 2mm within the range of test error. Figure 179 shows the simulation result of 2kg torsional load case with updated model. There are no noticeable differences as expected, because the parameter is updated according to the bending load case.

After model updating, a modal analysis is conducted with updated model. Figure 180 shows comparison of eigenfrequencies between GVT, FEM and updated FEM. Only the 3n asymmetric wing bending is improved, all other modes become worse. This is due to the fact that updating with the static test, the engineering stiffness (E_1) is tuned down. This results in tuned down eigenfrequencies. Next step is to use tuning beams locally to improve the 3n bending while not destroying the other mode shapes.

Comparison of RCE aircraft model with static test and GVT The initial model generated with MDO toolchain was prepared to replicate the static test set up. The results can be seen from figure 181 that the RCE model is way stiffer than the manufactured. Using the same approach as for -1 wing updating,

a knock-down factor was applied to the engineering stiffness of wing spar and skin. The figure 182 shows the wing bending results with updated RCE model. The wing bending is much closer to the static test results.

A modal analysis is also conducted for the initial RCE model and updated RCE model. The results can be seen in the figure 183. After the update, all the modes listed are improved.

Validation of the low order aeroservoelastic (ASE) model The last step is to validate the accuracy of the low order ASE model that is constructed in the modeling block of the RCE framework. This model serves as the basis for the automatic baseline and flutter suppression control design algorithms. The model is a set of linear time invariant (LTI) models that are obtained form the nonlinear model by Jacobian linearization at airspeed values between 38 and 64 m/s. The model needs to capture the low frequency dynamics of the aircraft for the baseline control design and the flutter modes for the flutter suppression design.

The base model is the low order model of the Flexop aircraft that is described in [70]. The pole map trajectories (as function of the airspeed) of the base model and the RCE generated models are shown in Figure 75.

The plots show good match between the legacy Flexop and the RCE generated model. The pole trajectories show similar trends and the interdependency between them is also very similar between the two modeling frameworks.

Aerodynamics Model Validation The Deliverable D3.2 – Flight Test Report Phase 1 described the taxi tests, flight tests and aerodynamic analysis performed within years 2020 and 2021. An issue is reported in the deliverable about the actual aircraft producing significantly less lift than was initially modelled. An almost constant lift coefficient offset of around 0.2 can be observed, which results in 35-45 percent lift loss in the 2-4 deg angle of attack region. FT5 and FT7 data do align in the same trend. Accordingly, an investigation was launched not only in the available flight test data, but also in aerodynamic modelling tools. The further findings from the flight test data are presented in the Deliverable 3.6 – Flight Test Report Phase 2.

In order to select an aerodynamic tool which models the aerodynamic characteristics of T-FLEX correctly, a comparison study of different aerodynamic tools was conducted.

In the mean time, MDO toolchain poses additional requirements on the tools regarding the simulation time and automatic execution.

Investigated tools includes the low order aerodynamic tools, Athena Vortex Lattice (AVL), XFLR5, PyTornado, Tornado, VSPAERO, PAWAT, FlightStream and high fidelity tool STAR-CCM+.

Aerodynamics modelling tools This section will give introduction to the investigated tools. For more details please refer to the paper [121].

AVL AVL is a program for performing aerodynamic analysis of rigid aircraft of arbitrary configurations [21]. It uses the VLM method to model the lifting surfaces. Because of an intrinsic limitation of VLM, AVL is only suitable for inviscid calculation at small angles of attack and sideslip.

Tornado Tornado is a Vortex Lattice Method for linear aerodynamic wing design applications in conceptual aircraft design or in aeronautical education [73]. The method is built in MATLAB [1] and is based

on the description as provided by Moran [78].

The VLM implementation in Tornado ignores the thickness effects of the airfoil, but includes the camber. In Tornado, modelling of control surfaces is possible. Experimental functions that generate a Trefftz-plane analysis can be used. Different options for mesh creation (linear panel distribution, cosine panel distribution, etc.) are available. A graphical interface is available which can plot coefficients of interest, display geometries and mesh.

Initially, Tornado was designed only to include linear aerodynamics. However, the code has been updated to include viscous effects as well [17].

If required, stability derivatives can be calculated using central-difference approximation around the trim condition. It is also possible to calculate trimmed polars.

PyTornado PyTornado is an aerodynamic tool for conceptual aircraft design. Short computation times make it possible to easily obtain estimates of aerodynamic loads and to benchmark different concepts [28]. Although a similar name as Tornado, PyTornado has been implemented from scratch within the European research project AGILE. A Vortex Lattice Method is implemented in this code. It has a user interface, pre- and post-processing in Python and a calculation core routine in C++ [74], which guarantees a user friendly interface and computational efficiency. It can be used as a standalone aerodynamic solver or can be integrated into a MDO toolchain. The deformation feature, which is under development, could be potentially used for aeroelastic analysis.

XFLR5 XFLR5 is a software tool designed specifically with model sailplanes in mind [18][19]. Therefore, it focuses on wings operating at low Reynolds numbers. The tool uses XFoil [22] (XFoil v6.99 since XFLR5 v6.55) to calculate the 2D aerodynamics of an airfoil. Non-linear Lifting Line Theory (based on the NACA technical note 1269 [99]), Vortex Lattice Method with quadrilateral rings (as recommended by Katz and Plotkin [45]) or 3D Panel Method (based on Maskew [69]) can be used for 3D wing and tail analysis. Body analysis is not recommended by the author [19].

Unlike the usual VLM solvers, the VLM method implemented in XFLR5 provides a viscous drag correction. In such case, lift-related characteristics (lift distribution, induced drag) are kept inviscid and after local lift distribution is calculated, viscous drag correction using 2D airfoil polars is applied. The lift distribution is not changed. This method is also used during this study. However, the author of the software raises awareness that such correction is not scientifically sound, as using 2D polars ignores any spanwise effects [19].

VSPAERO VSPAERO [51] is the aerodynamic analysis tool integrated within the conceptual aircraft design package OpenVSP [83]. The tool has two methods available - the Vortex Lattice Method with a simple stall prediction methodology (not used in this study) and a 3D Panel method [52]. Propellers can be included in the simulation. The tool also incorporates the possibility to calculate the parasite drag using the component build-up method. In the current study, only the VLM method is used.

PAWAT The Preliminary Design Tool for Propeller-Wing Aerodynamics (PAWAT) is an aerodynamic tool for the conceptual design of aircraft [102]. The calculation of the steady state lifting surface aerodynamics in PAWAT is based on a modified three-dimensional nonlinear lifting line theory with a fixed wake model employing nonlinear airfoil data to model nonlinear and viscous effects to a certain extent [102]. PAWAT is also capable of modelling propellers and it allows investigations of the interaction effects between wing and propeller.

The method is built in MATLAB [1]. The description of the lifting line method used is described by Phillips and Snyder [87].

FlightStream FlightStream is a novel surface vorticity solver capable of using structured or unstructured surface meshes. As a vorticity-based solver, the code can be expected to be substantially more robust and stable compared to pressure-based potential-flow solvers and less sensitive to surface perturbations, and it also allows the use of coarser meshes with an acceptable level of fidelity [80].

To account for viscous effect, integral boundary layer was implemented in FlightStream and was coupled with inviscid solver via displacement of the inviscid boundary equal to the displacement thicknesses of the local boundary layers. More features like prediction of flow separation and stall characteristics are also enabled by this implementation.

STAR-CCM+ Simcenter STAR-CCM+ is a multiphysics computational fluid dynamics (CFD) software. In this study, it is used to provide the reference data for comparison.

One has to emphasize that most of the tools simulated the wing and tail (Figure 76). Fuselage was included only in simulations of STAR-CCM+ and FlightStream. A study was done with STAR-CCM+ to investigate the influence of the fuselage on the spanwise lift distribution. A small influence was noted at low angles of attack. However, at high angles of attack the fuselage does change the flow at the wing root.

Global aerodynamic coefficients This section will compare the aerodynamic tools regarding the global aerodynamic coefficients. For more details please refer to the paper [121].

Lift The lift coefficient data is plotted with respect to the angle of attack in Figure 77 as well as in Figure 78 for the linear part of the slope. The lift curve slope coefficients C_{L_α} and zero angle lift coefficients C_{L_0} are shown in Table 6.

Table 6: Comparison of lift curve slope C_{L_α}, zero angle lift coefficient C_{L_0}, minimum drag coefficient $C_{D_{min}}$, pitching moment curve slope C_{m_α} and zero angle pitching moment coefficient C_{m_0} for different aerodynamic modelling tools.

	Turbulent	Euler	FlightStream	PAWAT	Tornado	AVL	XFLR5	VSPAERO	PyTornado
C_{L_α}	0.106	0.111	0.103	0.107	0.105	0.104	0.104	0.104	0.104
C_{L_0}	0.206	0.248	0.214	0.180	0.122	0.185	0.198	0.205	0.215
$C_{D_{min}}$	0.020	0.005	0.015	0.016	0.001	0.002	0.015	0.012	0.002
C_{m_α}	-0.027	-0.028	-0.030	-0.047	-0.050	-0.032	-0.032	-0.026	-0.030
C_{m_0}	0.132	0.141	0.117	0.103	0.193	0.214	0.147	0.128	0.159

Significant reduction in lift is apparent when comparing the turbulent simulations to Euler simulations. This is expected, as the viscous boundary layer on the top surface of the wing reduces the effective camber line, therefore reducing the aerodynamic angle of attack. Interestingly, most of the tools show better alignment with the turbulent simulations than with the inviscid ones, even though only PAWAT and FlightStream take viscosity into account when calculating lift.

When only the linear part of the lift curve is concerned, the calculated curve slope agreed with each other. The zero angle of attack lift shows deviations among tools. Tornado differs most from the other tools. Taking into account that all the tools are meant for preliminary design phase, the differences between them could be categorised as being insignificant.

The nonlinear part of the curve is predicted by both PAWAT and FlightStream. Even at high angle of attack, the lift curve from FlightStream matches quite good with CFD turbulent result. However, as no CFD simulations above 14 degrees were done, the $C_{L_{max}}$ could not be estimated.

The spanwise normalized lift distribution for $\alpha = 2\,\mathrm{deg}$ is plotted in Figure 79. As only the shape of the distribution is of importance here, the local lift coefficients are normalized with respect to the maximum local lift coefficient for the same tool.

The normalized lift distributions between the turbulent and Euler simulations are almost identical. The estimated maximum local lift location is similar for all the tools. The overall shape is very similar with some discrepancies at the root and tip areas. The differences between the STAR-CCM+ results and the other tool results at the wingtip might be due to the poor discretization when extracting the lift distribution from STAR-CCM+.

Drag Figure 80 shows the inviscid drag polar. While all the VLM tools and panel-based method FlightStream agree mainly, the differences compared with the STAR-CCM+ Euler simulation are noticeable even at low lift coefficient.

One has to note that the inviscid drag extracted from STAR-CCM+ here is the pressure drag component acting on the aircraft. Strictly speaking, this is not equal to the induced drag by definition. The separation of induced and profile drag from CFD is not straight-forward, and if Euler simulations are used the induced drag due to viscous effects are then ignored. Nowadays there exist some methods to extract these two drag components from CFD [62], but they were not implemented at the time of writing this report.

The total drag coefficient shown in Figure 81 includes both viscous and inviscid drag. Significant differences can be seen in between the tools that correct for viscous drag (STAR-CCM+ (turbulent), FlightStream, PAWAT, XFLR5, VSPAERO) and the ones that do not (Tornado, AVL, PyTornado).

Different methods were used to correct the viscous drag in different software tools. Variation of viscous drag is clearly visible in the Figure 81.

Both PAWAT and XFLR5 correct viscous drag based on 2D airfoil polar data. For XFLR5, 2d viscous drag is interpolated from local wing lift coefficient. The interactive boundary layer, which is a coupling method between potential flow and viscous flow on surfaces, is not implemented in the VLM available in XFLR5 [18]. The consequence of underestimation of viscous drag is confirmed in the Figure 81.

In PAWAT, equations are established for wing segments based on the aerodynamic force derived from three-dimensional vortex lifting law and the aerodynamic force derived from nonlinear airfoil characteristics of the segment and the segment area [102]. Iterative procedure is needed to solve the equations. Total drag coefficient from PAWAT matches quite well with CFD data.

In FlightStream, integral boundary layer is coupled with inviscid surface solver to account for viscous drag. Even though, the total drag seems to be underestimated.

Pitching moment The pitching moment coefficient with regard to angle of attack is shown in Figure 82. The results predicted by STAR-CCM+ Euler simulation and turbulent simulation are almost identical, except that at the high angle of attack, turbulent simulation shows a pitch up trend. FlightStream shows a similar pitch up trend as turbulent simulation, even though an offset of the curve is visible.

The pitching moment coefficients predicted by VSPAERO, XFLR5 and PyTornado match quite good with the CFD results. The results from PAWAT, AVL and Tornado show noticeable deviation to the reference.

Flight Dynamics Model Validation The flight dynamics of a flexible air vehicle is characterised by an aero(servo)elastic (ASE) model that considers the interaction of aerodynamics, structural dynamics, rigid body dynamics, and control laws which comprise interdisciplinary aircraft modelling. These subsystems can be independently modelled using a theoretical approach, and experimental results from various ground and flight tests can be incorporated into the models. Before flight tests can be conducted safely and effectively, a number of ground tests must be performed. The static and ground vibration tests (GVT) are essential for evaluating and improving the accuracy of the numerical models used during the aircraft design phase. The results of ground testing, for instance, are used to update and validate the Finite Element (FE) model that represents the structural dynamic part of the aircraft model. Similarly, an aerodynamic model can be generated using computational fluid dynamics (CFD) simulations or panel methods derived from potential theory, typically representing quasi-steady and unsteady aerodynamics, which can be partially updated using limited data from wind tunnel-tests, which are often used in the aircraft design and development phase for configuration optimization. Compared to completely flight mechanical models representing rigid body aircraft dynamics, which are adequately represented by low-order dynamics, and purely structural models, which are adequately represented by higher-order dynamics, ASE models encompass both the low- and high-frequency range.

For the validation of the flight dynamics model, an updating methodology for correcting of the numerical system matrices A, B, C and D of the linear discrete-time state-space models of the flexible aircraft has been developed combining the phenomenological and behavioral model structures. The proposed approach updates the system matrices using the measured input-output data from flight test and the initial state-space model of the flexible aircraft which is derived from the linearization of the nonlinear first-order differential equations describing the aircraft motion. The key feature of the proposed approach is that updating can be executed in a single step with multiple data bases from different flight tests with nearly-identical initial conditions, resulting in a more physically realistic correction of the system matrices. The updating method addresses linear estimation problems which allows an manageable implementation with fast execution avoiding optimization problems for approximation of solutions of nonlinear differential equations resulting from aircraft equations of motion.
The primary purpose of numerical model improvement is to identify and correct (update) the discrepancies between experimental and predicted numerical outputs. In the case of significant disparities between model predictions and experimental data from flight tests, the numerical aircraft model must be updated until there is a satisfactory correlation between model predictions and experimental results.

The proposed updating algorithm and its advanced application is based on the study described in [104]. The formulation of the proposed updating approach enables correction of the system matrices A, B, C, and D of the initial linearized discrete-time (DT) state-space system derived from a flexible aircraft model. The updating method addresses linear estimation problems combining the phenomenological and behavioral model assumptions.

Thus, two formulations for the error minimization, i.e. minimization of the output residual between flight measured data and model predictions have been defined. The first utilises the state-space system's output equations, whereas the second requires both the state and output equations. The methodology for updating that will be described here is based on a linear least-squares approximation resp. a minimum norm solution. The updating algorithm has three steps. In the first step, the calculated states from the initial model corresponding to the rigid body aircraft dynamics will be corrected using output equations. Here, the considered states are measured and comprise a subset of the outputs. In the second step, we use the same approach as in the first. Here, the principal difference is that we consider the states corresponding to rigid body and flexible aircraft dynamics. In the third step of the algorithm,

the system matrices A, B, C, and D are directly updated using the updated states from the previous two steps. Note that the algorithm described in [104] consists of four steps. Here, we omit the fourth step in which the system matrices C and D are reestimated to ensure a better output match without regard to the actual system's internal behaviour. Within the context of the method and model validation analysis used for this study, we evaluate the quality of the updated model by residual analysis using Theil's inequality for assessment of the fit. Theil's inequality and the breakdown of fit error in terms of bias, variance and covariance proportions, offer insight on the validity of the predicted responses from the updated model. The following diagram (Fig.375) illustrates a summary of the study presented in this work, including the flight test domain, model structure, updating algorithm, and model validation process.

Updating algorithm This section outlines the formulation of the proposed updating approach for the numerical system matrices $\mathbf{A}_d$, $\mathbf{B}_d$, $\mathbf{C}_d$, and $\mathbf{D}_d$ of the discrete-time linear state-space system representing the flexible aircraft model. The main objective of numerical model improvement is to detect and correct the discrepancies between experimental and estimated system outputs. The updating methodology is based on error minimization of the output residuals using both state and output equations. Its algorithm is a three-step procedure and it is based on a linear least-squares approximation resp. a minimum norm solution. We begin by defining the error minimization formulations used in the updating approach, and then we outline the related updating steps.

The first error minimization problem which represents an output residual formulation between flight test data and model predictions by using only the output equations, is defined as:

$$\min_{\Delta x_{k,i}} \sum_{i=1}^{N_{test}} \sum_{k=1}^{N_t-1} \|\overline{y}_{k,i} - \mathbf{S}_{sens} \cdot (\mathbf{C}_{\mathrm{d}} \Delta x_{k,i} + \mathbf{D}_{\mathrm{d}} (\overline{u}_{k,i} - u_{0,i}) + y_{0,i})\|^2 \,. \tag{2}$$

We shall denote the value of perturbed state vector $\Delta x_{k,i}$ that minimizes Eq. (70) by $\Delta x_{k,i}^{(c)}$, for fixed $\mathbf{C}_d$ and $\mathbf{D}_d$:

$$\Delta x_{k,i}^{(c)} \;=\; \arg\min_{\Delta x_{k,i}} \sum_{i=1}^{N_{\mathrm{test}}} \sum_{k=1}^{N_t-1} \|\overline{y}_{k,i} - \mathbf{S}_{sens} \cdot (\mathbf{C}_d \Delta x_{k,i} + \mathbf{D}_d (\overline{u}_{k,i} - u_{0,i}) + y_{0,i})\|^2 \,, \tag{3}$$

where $\overline{u}_{k,i} \in \mathbb{R}^m$ and $\overline{y}_{k,i} \in \mathbb{R}^{\bar{l}}$ are measured inputs and outputs from ith flight test. N_{test} denotes the number of flight test sets used for model updating. For clarification: $\bar{l}$ is the number of outputs from test and l is the number of outputs from numerical model. $\mathbf{S}_{sens} \in \mathbb{R}^{\bar{l} \times l}$ is the sensor matrix allocating the measured outputs with the estimated outputs from model. It is defined as an identity matrix $\mathbf{S}_{sens} = \mathbf{I} \in \mathbb{R}^{l \times l}$ if all outputs from the numerical model are measured. Otherwise it becomes a rectangular matrix $\bar{l} < l$. For most cases, a sensor matrix is needed when you are interested in outputs that are essential but cannot be measured during the test, or when only a subset of measured quantities is intended for model updating. The error minimization formulation given in 70 is used for the first and second step of the updating algorithm, where for fixed $\mathbf{C}_d$ and $\mathbf{D}_d$ the perturbed states Δx_k can be corrected.

The second error minimization formulation requires both state and output equations where system matrices $\mathbf{A}_d$, $\mathbf{B}_{\mathrm{d}}$, $\mathbf{C}_{\mathrm{d}}$ and $\mathbf{D}_{\mathrm{d}}$ can be corrected by using the updated states $\Delta x_{k,i}^{(c)}$ obtained from the Eq.

(71):

$$\mathbf{A}_d^{(c)}, \mathbf{B}_\mathrm{d}^{(c)}, \mathbf{C}_d^{(c)}, \mathbf{D}_d^{(c)} = \arg \min_{\left(\begin{smallmatrix} A_d & B_d \\ C_d & D_d \end{smallmatrix}\right)} \sum_{i=1}^{N_{test}} \sum_{k=0}^{N_t-2} \left\| \begin{pmatrix} \Delta x_{k+1,i}^{(c)} \\ \overline{y}_{k,i} \end{pmatrix} - \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{S}_{sens} \end{bmatrix} \left(\begin{bmatrix} \mathbf{A}_d & \mathbf{B}_d \\ \mathbf{C}_d & \mathbf{D}_d \end{bmatrix} \begin{pmatrix} \Delta x_{k,i}^{(c)} \\ \overline{u}_{k,i} - u_{0,i} \end{pmatrix} + \begin{pmatrix} 0 \\ y_{0,i} \end{pmatrix} \right) \right\|^2 . \tag{4}$$

Here, the error minimization formulation in Eq. (72) is used in the third step of the updating algorithm. The three-step procedure of the proposed updating method may be summarised as follows:

1. Correction of measured states corresponding to rigid body aircraft dynamics using the first error minimization formulation given in Eq. (71)

2. Correction of measured and unmeasured states relating to rigid body and flexible aircraft dynamics using the first formulation for error minimization given in Eq. (71)

3. Correction of system matrices $\mathbf{A}_d$, $\mathbf{B}_d$, $\mathbf{C}_d$ and $\mathbf{D}_d$ using the updated states from the first and second step using the second formulation for error minimization given in Eq. (72)

Detailed mathematical derivation of the updating methodology can be found in [130].

Model Validation Facts are distinct from estimates. Model validation is essential for gaining confidence in or rejecting a certain model. Comparing measured and simulated outputs is required to validate the updated model. There are numerous aspects of model validation that can be broadly categorised into three subcategories [42]:

- Statistical properties of the estimates

- Residual analysis

- Model predictive quality

These three methods offer insight into the effectiveness or ineffectiveness of model parameters. They provide the essential means to evaluate the suitability of identified (updated) models and their parameters in duplicating the system closely enough.

Within the framework of the method and model validation analysis employed in this study, we evaluate the quality of the updated model by residual analysis using Theil's inequality for assessment of the fit. The prediction capability of the updated model, which will be described later in this section, is then evaluated.

Theil's inequality analysis and decomposition of fit error Theil's inequality provides an usefull type of output statistics for the overall fit. Theil's inequality coefficient TIC is defined as

$$\mathrm{TIC_o} = \frac{\sqrt{\frac{1}{N_t}\sum_{k=0}^{N_t-1}\left(\overline{y}_{k,o} - y_{k,o}\right)^2}}{\sqrt{\frac{1}{N_t}\sum_{k=0}^{N_t-1}\left(\overline{y}_{k,o}\right)^2} + \sqrt{\frac{1}{N_t}\sum_{k=0}^{N_t-1}\left(y_{k,o}\right)^2}}, \quad o = 1, 2, ..., \bar{l} \tag{5}$$

where $y_{k,o}$ is the o'th output from model at time $t = kT_s$ and $\overline{y}_{k,o}$ is its counterpart from flight test. Theil's inequality coefficient measures the conformity between two time series. In statistical terms, it is the ratio of the root-mean-square fit error to the sum of the root-mean-square values of the measured and estimated signals. TIC $= 0$ (case of equality) implies a perfect match, whereas TIC $= 1$ indicates the case of maximal inequality.
Additionally, Theil analysed the fit error between the two time series in terms of bias, variance, and covariance proportions given by [65], [112]:

$$\mathrm{TIC_{M,o}} = \frac{\left(\overline{y}^M_{k,o} - y^M_{k,o}\right)^2}{\frac{1}{N_t}\sum_{k=0}^{N_t-1}\left(\overline{y}_{k,o} - y_{k,o}\right)^2} \tag{6}$$

$$\mathrm{TIC_{S,o}} = \frac{\left(\sigma_{\overline{y},o} - \sigma_{y,o}\right)^2}{\frac{1}{N_t}\sum_{k=0}^{N_t-1}\left(\overline{y}_{k,o} - y_{k,o}\right)^2} \tag{7}$$

$$\mathrm{TIC_{C,o}} = \frac{2\left(1-\rho_o\right)\sigma_{\overline{y},o}\sigma_{y,o}}{\frac{1}{N_t}\sum_{k=0}^{N_t-1}\left(\overline{y}_{k,o} - y_{k,o}\right)^2} \tag{8}$$

wehre $\overline{y}^M_{k,o}$ and $y^M_{k,o}$ refer to the mean values of the oth measured and simulated output. σ and ρ are the standard deviations and correlation coefficient respectively of the two output signals $\overline{y}$ and y. They are defined as

$$\sigma_{\overline{y},o} = \sqrt{\frac{1}{N_t}\sum_{k=0}^{N_t-1}\left(\overline{y}_{k,o} - \overline{y}^M_{k,o}\right)^2} \quad , \quad \sigma_{y,o} = \sqrt{\frac{1}{N_t}\sum_{k=0}^{N_t-1}\left(y_{k,o} - y^M_{k,o}\right)^2} \tag{9}$$

$$\rho_o = \frac{1}{\sigma_{\overline{y},o}\sigma_{y,o}}\frac{1}{N_t}\sum_{k=0}^{N_t-1}\left(\overline{y}_{k,o} - \overline{y}^M_{k,o}\right)\left(y_{k,o} - y^M_{k,o}\right) . \tag{10}$$

Again, we separate the three proportions defined in Eqs. 6-8 for each output.

The bias proportion $\mathrm{TIC_{M,o}}$ is a measure of the systematic error in the updated model, while variance proportion $\mathrm{TIC_{S,o}}$ indicates the model's capacity to duplicate the variability of the true system. Nonsystematic error is quantified using the covariance proportion $\mathrm{TIC_{C,o}}$. The above breakdown offers insight into the sources of fit error. For an ideal case, the bias and variance proportions should be close to zero, and the covariance proportion should be close 1. The sum of these three proportions equals 1. For both $\mathrm{TIC_{M,o}}$ and $\mathrm{TIC_{S,o}}$, a large value, often greater than 0.1, would be cause for concern and the updated (identified) model should be scrutinised and analysed in detail. In conjunction with a visual evaluation of the fit between the output signals, these criteria give slightly more insight into the characteristics of residuals [130].

Model predictive capability The predictive capability check of the updated model is determined by comparing the flight measured responses with those estimated by the updated model for the ”same” control inputs. This requires a flight test data set with the nearly identical control inputs and trim conditions as chosen in identification tests for model updating. In the terminology of aircraft applications, this procedure is frequently referred to proof-of-match (POM), which is not an easy task. Both the control input and output measurements are susceptible to measurement noise. In addition, even when the proof-of-match manoeuvres can be performed in apparently calm atmospheric conditions, the aircraft is excited by a small amount of nonmeasurable turbulence-induced excitation. In general, complementary flight data, i.e. flight manoeuvres not used in the identification tests for model updating, are used to evaluate the model predictive capability. Validation on complementary data is sometimes referred to informally as a ”acid test” [43].

Case Study The algoritm of the proposed updating method is coded in MATLAB. With both the discrete state-space model and access to flight test data of the FLiPASED demonstrator aircraft, the application of the proposed updating methodology is demonstrated, followed by the presentation of convincing findings.

Flight Test The FLiPASED aircraft is a jet-powered UAV with a wing span of 7.1 m and a takeoff mass of 65 kg. It was manually operated within visual line of sight. Ref.[120] offers more insight into the flight test campaign. The FLiPASED aircraft is equipped with an integrated measurement system that is considered in both the nonlinear aircraft model and the linearized state-space representation, respectively. The usual air data, position and inertial parameters are being logged on the aircraft. Attached to the front and rear spars are 12 inertial measurement units (IMU) that records the structural deflections of the wings. The wing-mounted IMUs measure translational accelerations in z direction and the angular rates ω_x and ω_y. As outputs, 36 time histories are therefore provided. Figure 376 shows the configuration of the IMUs' placement on the wings [103]. Further, a fuselage-mounted IMU provides flight measured translational accelerations a_{Fuse} and the rotational rates Ω_{Fuse}. More information on flight test instrumentation (FTI) is given in Ref. [120].

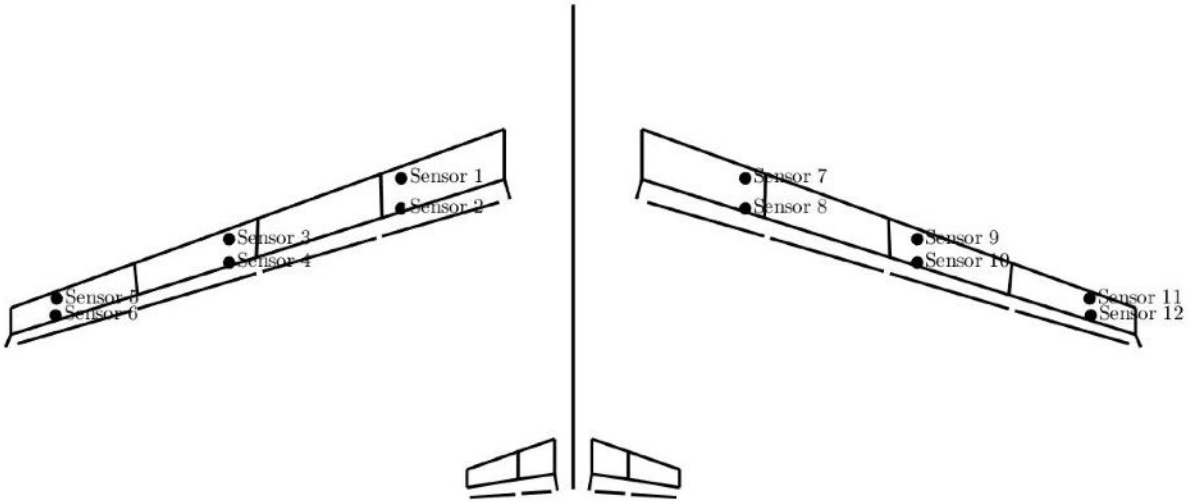


Figure 84: Location of accelerometers (IMUs) on FLiPASED aircraft [104]

The flight test data used in this study are provided by a pushover-pull-up manoeuvers. The primary objective of pushover–pull-ups, commonly known as roller-coaster, is to identify lift and drag characteristics, longitudinal stability, and elevator trim requirements. The maneuver starts from a trimmed level flight condition with a constant thrust [42]. With a sampling rate of 200 Hz and a 20-second time window, experimental data from three test sets have been used for model updarting method. The recorded input/output time series are then upsampled to 1 kHz to obtain data consistency with the discrete state-space model of the aircraft. The flight measured outputs with their physical quantities are listed in the table 7. In addition, the trim conditions obtained from flight test measurements are stated in table below (Tab.82).

Table 7: Physical quantities of the states and inputs-outputs of the linear discrete-time state-space system of the aircraft model*

States				Inputs	Outputs
kinematic	dynamic	elastic	aerodyn. lag		
$\Delta\phi$	Δu	$\Delta\eta_f$ (30)	Δx_L (288)	$\Delta\xi$ aileron defl. (8)	ϕ
$\Delta\theta$	Δv	$\Delta\dot{\eta}_f$ (30)		$\Delta\eta$ elevator defl.(4)	θ
$\Delta\psi$	Δw			$\Delta\delta_F$ Thrust setting	ψ
Δx_{0E}	Δp				$a_{x,\ \text{IMU-Fuse}}$
Δy_{0E}	Δq				$a_{y,\ \text{IMU-Fuse}}$
Δz_{0E}	Δr				$a_{z,\ \text{IMU-Fuse}}$
					$p_{\ \text{IMU-Fuse}}$
					$q_{\ \text{IMU-Fuse}}$
					$r_{\ \text{IMU-Fuse}}$
					z_E
					V_{IAS}
					α
					β
					h_{baro}
					p_{stat}
					p_{total}
					IMUs-Wing: a_z, ω_x, ω_y (36)

* The numbers in parentheses denote the number of corresponding physical quantity.

Table 8: Trim values measured from identification flight tests

	$\eta_{elev,0}$ [deg]	θ_0 [deg]	$V_{ias,0}$	$h_{baro,0}$ [m]	α_0 [deg]	β_0 [deg]
Identification Test #1	-3.68	0.99	38.5	814	2.78	0.403
Identification Test #2	-3.52	8.12	35.7	807	3.73	-0.818
Identification Test #3	-3.52	1.79	39.7	738	1.67	1.424

As is clear from the table above (82), the initial conditions assessed by three sets of flight data are not ”equal” as expected under real conditions. Here, we limit ourselves to the barometric altitude $h_{baro,0}$ and $V_{ias,0}$ parameters from which, inter alia, the generated state-space models are dependent.

For pushover-pull-up manoeuvres the aircraft is excited by elevator deflections, as depicted in Fig.378.

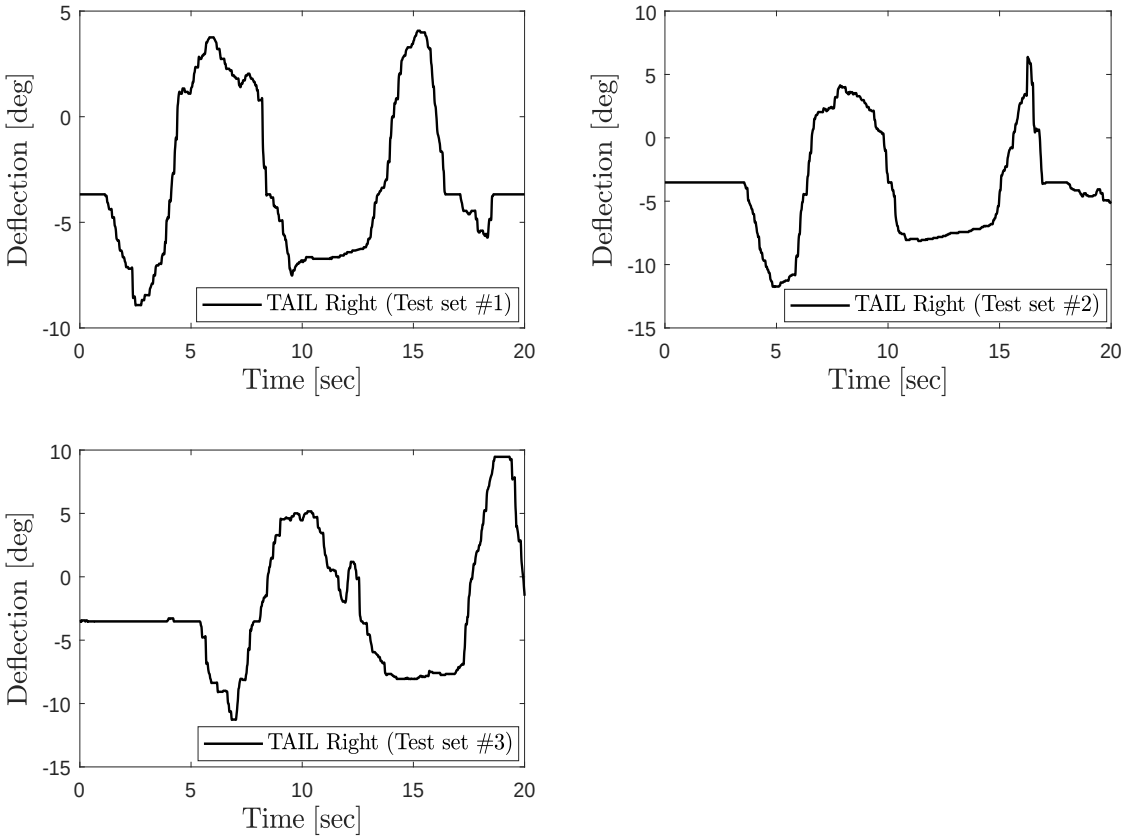


Figure 85: Elevator deflections used for pushover–pull-ups

Results The presented updating method has been successfully applied on the linearized FLiPASED aircraft model including the use of the flight test data. The results are based on the research described in [130].

Using residual analysis by means of Theil's inequality formulation given in 5, we assess the quality of the updated model for each of the output variables $o = 1, 2, ..., \bar{l}$. Although the acceptable value for TIC_o varies on the application, as a general guideline, a value < 0.25 indicates a satisfactory agreement. It is also feasible to establish a single measure for the overall fit. Thus, we define the mean Theil's inequality coefficient TIC_{mean} given by

$$\text{TIC}_{\text{mean}} = \frac{1}{\bar{l}} \cdot \sum_{o=1}^{\bar{l}} \text{TIC}_o \quad \% \tag{11}$$

Figure 86 illustrates the correlation results calculated from Theil's inequality formulation between the considered subset of output signals from three flight test sets and corresponding outputs from the initial and updated model.

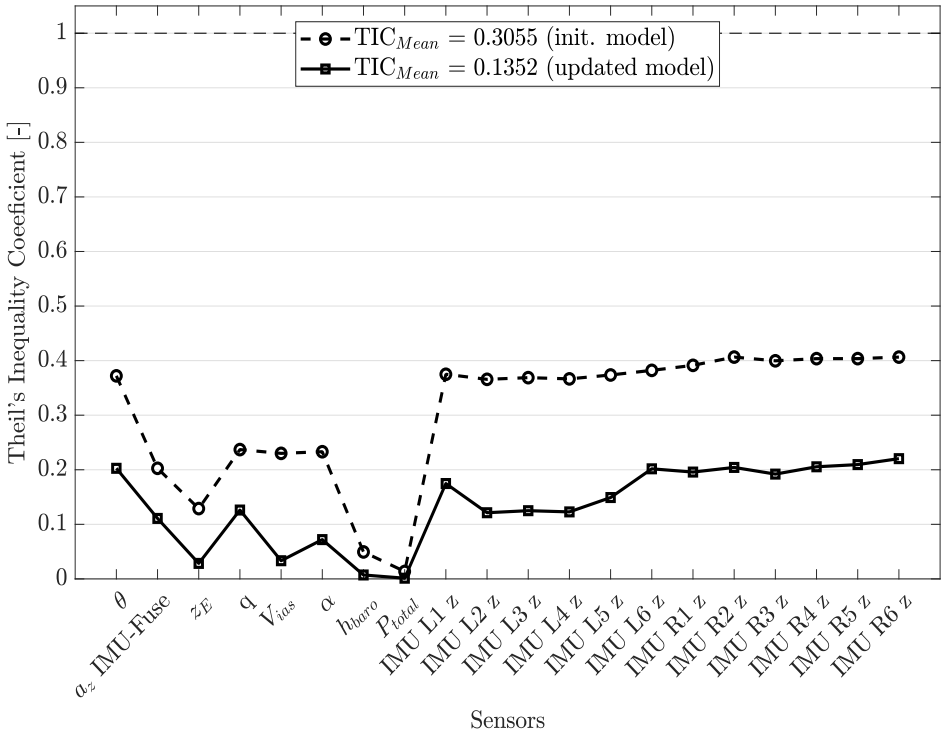


Figure 86: Fit error distribution between flight test and updated model data for each output (Number of test sets = 3)

As is evident from the plot in Fig. 86, a high degree of fit between flight test data and results from updated linearized aircraft model has been achieved. The mean Theil's inequality coefficient TIC_{mean} (Eq.11) between the outputs from updated model and the recorded data from flight test is approximately 0.14 and has decreased by 0.17. Again, $\text{TIC} = 0$ implies a perfect match (best fit), whereas $\text{TIC} = 1$ indicates the case of minimum correlation. In addition, we partitioned the fit error TIC_o for each output between the flight measured and reconstructed responses from the updated model into proportions of bias, variance, and covariance given in the equations 6-8 as shown in Fig. 386.

As already mentioned in Section 2.1.3, the bias and variance proportions TIC_M and TIC_S should be very small, typically $(TIC_M + TIC_S) \leq 0.2$. Again, TIC_M is an indicator of the systematic error in the updated model, while variance proportion TIC_S implies the model's capability to duplicate the variability of the physical system [42]. The diagram in Fig. 386 clearly demonstrates that nonsystematic error, represented by the covariance proportion TIC_C, predominates the source of the fit error with $TIC_{C,o} > 0.90$, $\forall o = 1, 2, ..., \bar{l}$, which suggests a high quality of the updated model, that is, capacity to duplicate the true system response.

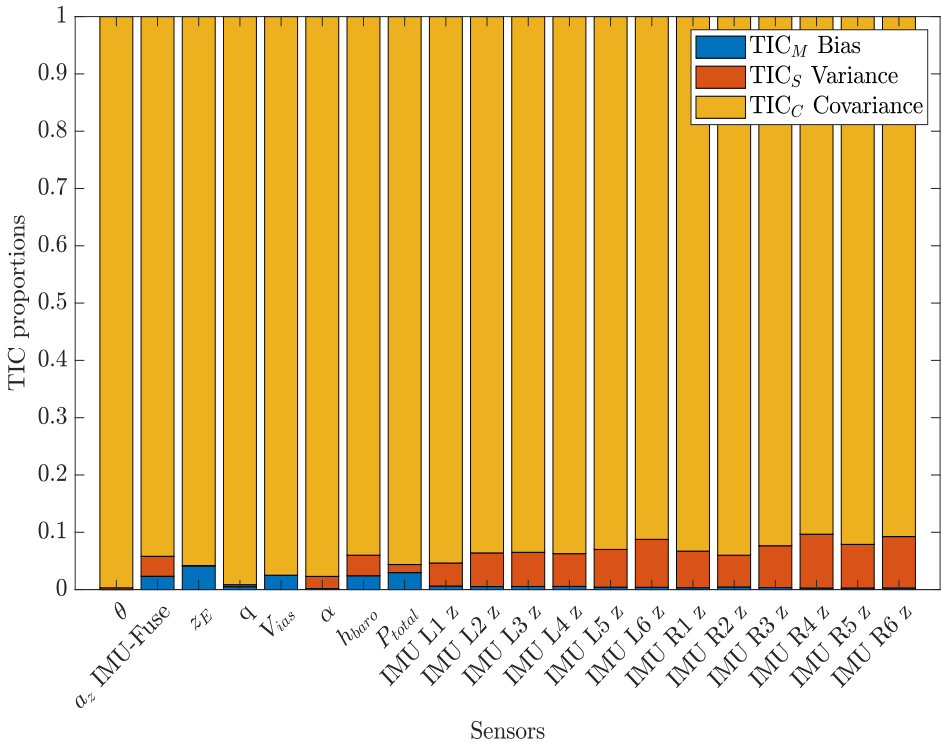


Figure 87: Breakdown of the fit error into proportions of bias, variance, and covariance (Number of test sets = 3)

Table (9) provides a detailed summary of the results obtained from Theil's inequality formulation including the decomposition of fit error between flight measured and reconstructed outputs from the updated model.

Table 9: Correlation results including the decompositon of fit error between flight measured outputs from identification tests and reconstructed outputs from updated model

	TIC [-]	TIC_M [-]	TIC_S [-]	TIC_C [-]
θ	0.20267	4e-05	0.003	0.99695
$az_{\text{IMU}-\text{Fuse}}$	0.11085	0.02322	0.03488	0.9419
z_{E}	0.02841	0.04142	0.00059	0.95799
q	0.12621	0.00506	0.00346	0.99149
V_{ias}	0.03331	0.0249	0.00016	0.97494
α	0.07202	0.00177	0.02124	0.97699
h_{baro}	0.00721	0.02412	0.03584	0.94003
P_{total}	0.00128	0.02959	0.01411	0.9563
IMU L1 z	0.17478	0.00614	0.04021	0.95365
IMU L2 z	0.12126	0.00526	0.05868	0.93606
IMU L3 z	0.12501	0.00539	0.05964	0.93497
IMU L4 z	0.1228	0.00553	0.05709	0.93738
IMU L5 z	0.1491	0.00417	0.06588	0.92995
IMU L6 z	0.20174	0.00398	0.08365	0.91237
IMU R1 z	0.19578	0.00325	0.06382	0.93293
IMU R2 z	0.20428	0.00452	0.05543	0.94005
IMU R3 z	0.19206	0.00347	0.07302	0.92351
IMU R4 z	0.20545	0.0027	0.09392	0.90338
IMU R5 z	0.20948	0.0028	0.07607	0.92113
IMU R6 z	0.22037	0.00282	0.08966	0.90752

In the following, a few selected outputs from the flight test are plotted together with the outputs from initial and updated linearized model of the aircraft (Fig. 380 - 384). It is clearly evident that the presented updating method enables to reconstruct all the measured responses obtained from the flight test with high accuracy even in case of highly noise-contaminated experimental data.

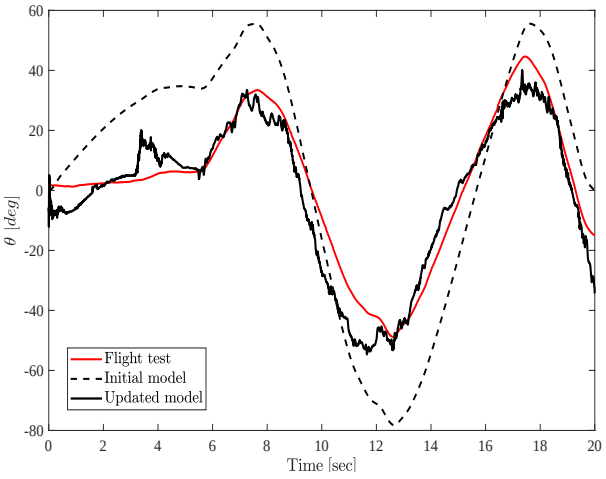


Figure 88: Pitch angle θ

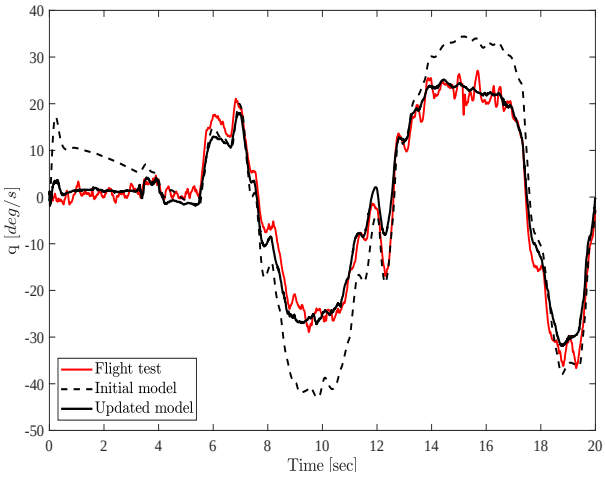


Figure 89: Pitch rate (q IMU-Fuse)

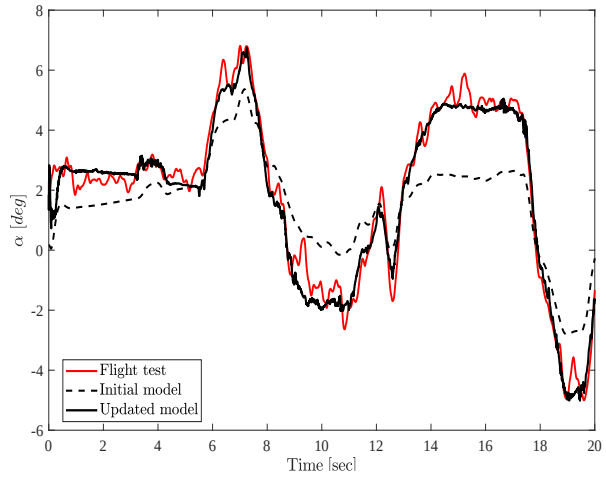


Figure 90: Angle of attack α

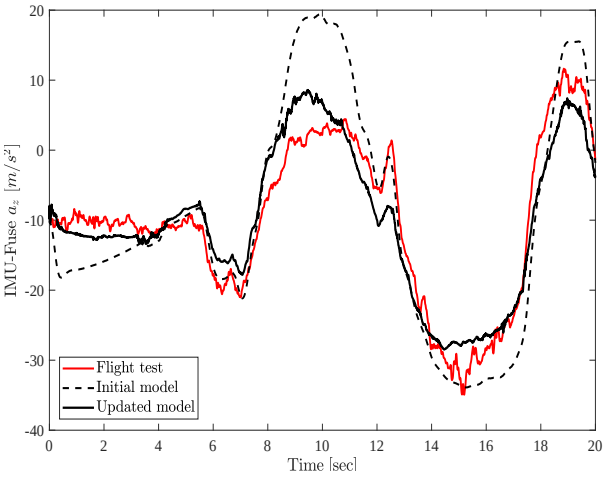


Figure 91: Vertical acc. $a_{z,\ \text{IMU-Fuse}}$

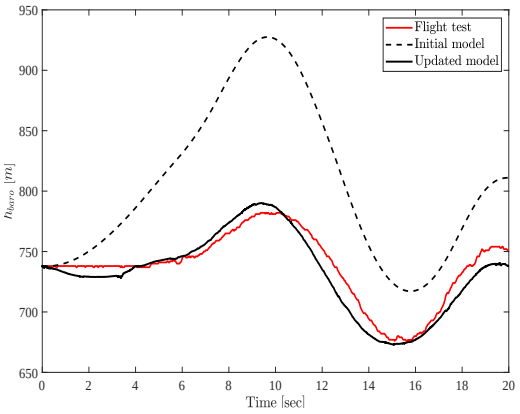


Figure 92: Barometric altitude h_{baro}

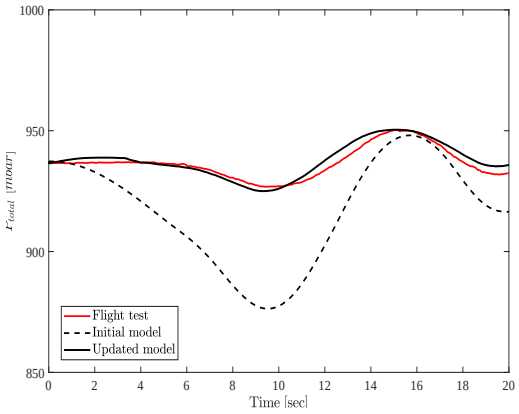


Figure 93: Total pressure P_{total}

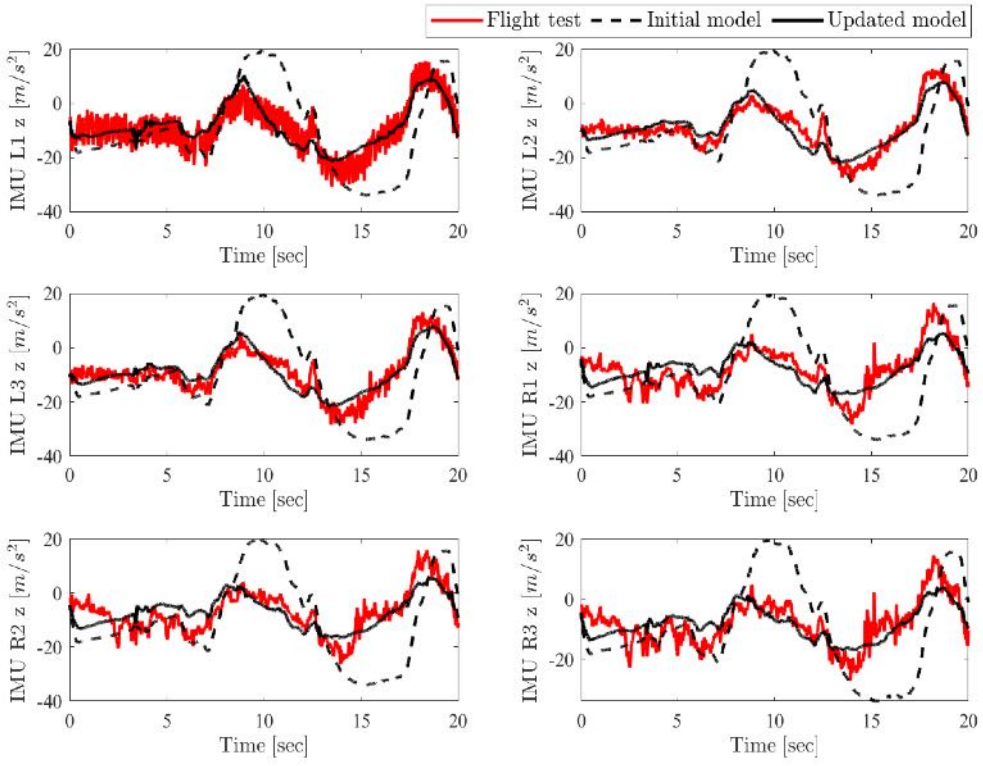


Figure 94: A subset of vertical accelerations $a_{\text{z,IMU}}$ recorded by six IMUs on the wings

Proof-of-match For model validation within the framework of proof-of-match procedure, a flight test data set with nearly identical control inputs and trim conditions is needed as chosen in identification tests for model updating. Hence, a suitable set of test data is chosen for model validation, where the aircraft is excited for a pushover-pull-up manoeuvre by an elevator deflection shown in Fig. (95) below.

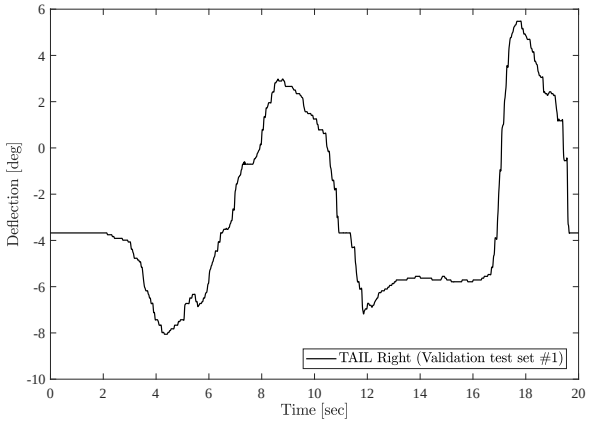


Figure 95: Measured elevator deflections and trim values from flight test for model validation

Trim values	
$\eta_{\text{elev},0}$ [deg]	-3.68
θ_0 [deg]	2.89
$V_{\text{ias},0}$	37.8
$h_{\text{baro},0}$ [m]	809
α_0 [deg]	2.64
β_0 [deg]	0.873

Figure (385) demonstrates that a high degree of match has been achieved between outputs from validation flight test and outputs estimated from the updated aircraft model. The mean Theil's inequality coefficient TIC_{mean} is approximately 0.18 and has decreased by 0.15.

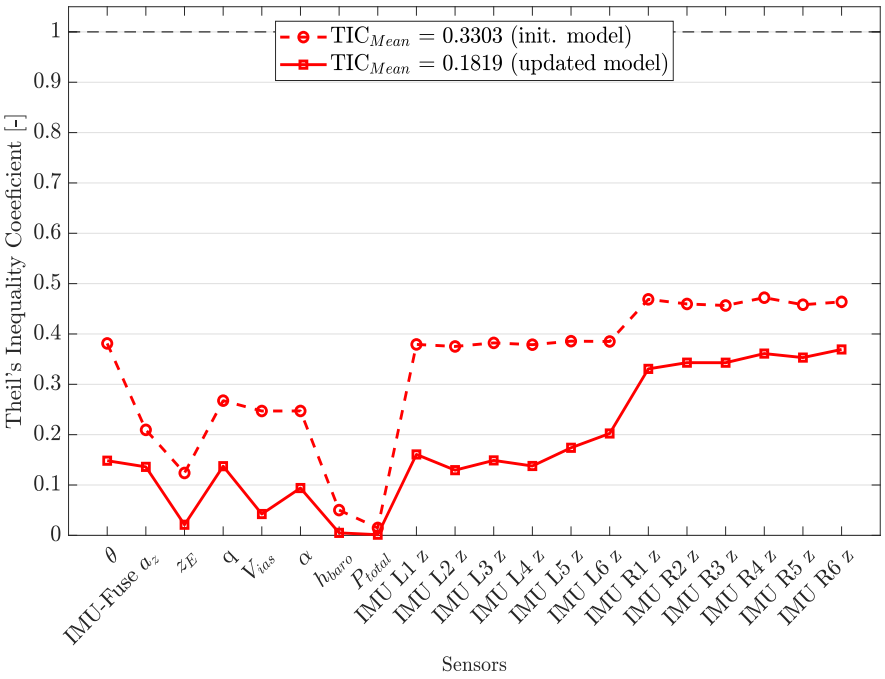


Figure 96: Fit error distribution between outputs from validation test and outputs provided from updated model for proof-of-match procedure

Conclusion In this study, it was proved that the proposed updating method can be successfully applied to discrete-time LTI state-space models describing rigid body and flexible aircraft dynamics by employing multiple flight test data. The suggested method permits adjustment of the predicted system matrices $\mathbf{A}_d$, $\mathbf{B}_d$, $\mathbf{C}_d$ and $\mathbf{D}_d$, beginning with the initial model, by minimizing the output residuals using both state and output equations. The methodology enables the reconstruction of all flight-measured responses with a high degree of overall correlation. The updated model has both phenomenological and behavioural characteristics, with the structure, sparsity, and density degree of the updated matrices remaining quasi-constant. The algorithm of the updating method described here is a three-step procedure and is based on linear least-squares approximation resp. a minimum norm solution which enables an appropriate implementation with fast execution avoiding optimization problems for approximation of solutions of nonlinear differential equations derived from aircraft equations of motion. Another benefit of the presented approach is that updating algorithm can be performed with different data bases derived from flight tests with the nearly identical initial conditions, which would lead to a more physically realistic correction of the system matrices. For the method and model validation, Theil's inequality formulation was utilised. This enables insight into the sources of fit error and hence assesses the quality of the updated model with physical insight.

Control System Design and Performance Validation The lead responsible partner for control system design is ONERA, but both DLR-SR and SZTAKI provided significant contribution to this task.

The present chapter discusses the following topics:

- Short summary of the flight control design tools for baseline, load and flutter control design
- Description of hardware-in-the-loop test results with control laws
- Description of flight test results with baseline control laws
- Description of used tools and how the model based design tools are validated to standardize them in the MDO toolchain

Baseline control structure The function of the baseline controller is to control the rigid body motion of the T-Flex aircraft. For this purpose a structured controller configuration was selected, which allows the sequential testing and tuning of the different control laws. The structure can be depicted in Figure 149.

The baseline controller has three operational mode:

(i) **Direct Mode**: The direct mode allows the pilot on the ground to bypass the flight control system. The only part active in the flight control computer is the mapping from the received remote-control signals to the commanded surface deflections. The pilot controls the pitch, roll and yaw axis directly via the aircraft's control surface deflections and its velocity via the thrust setting.

(ii) **Stability Augmentation Mode**: The augmented mode switches on basic augmentation for the pilot. Instead of directly controlling the surfaces the pilot inputs pitch- and roll-attitude commands. The side-slip angle is automatically regulated to zero, reducing the pilots need to control the yaw axis separately. Velocity control remains in direct control, i.e., the pilot controls the velocity via the thrust setting.

(iii) **Autopilot Mode**: In this mode the pilot fully delegates the aircraft control to the flight control system. Altitude, course angle, velocity and side-slip angle are automatically controlled. To fly along the defined test pattern, reference commands based on the aircraft position are generated in a navigation module.

The inner loops of the control system in roll, pitch and yaw provide the basis for the operational model (ii) and (iii). Mode (iii) is the core element of the autopilot adding the outer loops for course angle, altitude and speed control (autothrottle) as illustrated in Figure 149. Thus, a series of cascaded control loops is used to facilitate the control design task. As the cross-coupling between longitudinal and lateral axis is negligible, longitudinal and lateral control design is separated. Thrust commands δ_{th} which are transferred to an engine revolution command δ_ω via a nonlinear mapping and the elevator δ_e are the available actuators for longitudinal control.

Baseline control design Lateral-directional control generates aileron (δ_a) and rudder commands (δ_r), which is a multivariable problem and requires the coordinated use of aileron command δ_a and rudder command δ_r. The most inner loop features roll-attitude (Φ) tracking, roll-damping augmentation via the roll rate (p), and coordinated turn capabilities, i.e. turns without side-slip, via feedback of the side-slip angle (β). The outer loop establishes control of the course angle (χ). All controllers are scheduled with velocity to increase performance over the velocity range. Within the fully automated flight mode (iii) the reference signals for the velocity (V_{ref}), altitude (H_{ref}), and course angle (χ_{ref}) are provided by a dedicated navigation algorithm. It uses the GPS longitudinal and lateral position of the aircraft (x_a and y_a) as well as the current course angle (χ) to provide the commands.

Structure wise, the control loops use scheduled elements of proportional-integral-derivative (PID) controller structures with additional roll-offs in the inner loops to ensure that no aeroelastic mode is excited

by the baseline controller. A scheduling in dependence of the indicated airspeed V_{ias} is used to ensure an adequate performance over the velocity range from 32 m/s to 70 m/s. For the scheduling a first or second order polynomial in V_{ias} is applied. As an example the proportional gain $k_p = z_0 + z_1 V_{ias} + z_2 V_{ias}^2$ is depending quadratically on V_{ias} with the free parameters z_0, z_1, and z_2. A comprehensive summary of the used controller structures for each cascaded loop is provided in Table 10, including the channel description in the controller architecture and the implemented scheduling.

Table 10: Summary of the control loops of the FLEXOP baseline flight control system with the inner loop functions (first part) and autopilot functions (second part).

Control Loop	Channel	Structure	Scheduling
Pitch Attitude Control	$(\Theta_{ref} - \Theta) \rightarrow \delta_e$	PI	2nd-order polyn. in V_{ias}
Pitch Damping	$q \rightarrow \delta_e$	P	1st-order polyn in V_{ias}
Roll Attitude Control	$(\Phi_{ref} - \Phi) \rightarrow \delta_a$	P	1st-order polyn in V_{ias}
Roll Damping	$p \rightarrow \delta_a$	P	1st-order polyn. in V_{ias}
Yaw Control	$\beta \rightarrow \delta_r$	PID	2nd-order polyn. in V_{ias}
Autothrottle	$(V_{ref} - V_{ias}) \rightarrow \delta_{th}$	2 DOF-PID	none
Altitude	$(H_{ref} - H) \rightarrow \Theta_{ref}$	PI	2nd-order polyn. in V_{ias}
Course Angle	$(\chi_{ref} - \chi) \rightarrow \Phi_{ref}$	PID	2nd-order polyn. in V_{ias}

Note that the controller outputs δ_e, δ_a, and δ_r defer from the actual surface inputs to ease the actual control design task. Thus, they need to be transformed to physical actuator commands via an adequate control allocation. The T-Flex aircraft has multiple control surfaces and features combined rudder and elevator surfaces (ruddervators). The commands to the actuators of the two aileron pairs are determined by

$$\begin{array}{ccccc} \delta_{a,l2} & = & \delta_{a,l3} & = & 0.5\delta_a \\ \delta_{a,r2} & = & \delta_{a,r3} & = & -0.5\delta_a \end{array} \tag{12}$$

to generate the required differential aileron deflections for roll motion control. For the ruddervators superposition of the elevator command δ_e and the rudder command δ_r is applied by

$$\begin{array}{ccccc} \delta_{elev,l1} & = & \delta_{elev,l2} & = & \delta_e + 0.5\delta_r \\ \delta_{elev,r1} & = & \delta_{elev,r2} & = & \delta_e - 0.5\delta_r. \end{array} \tag{13}$$

Thus, symmetric deflections on the left and right of the ruddervators correspond to elevator commands while differential deflections establish rudder commands.

Parameter Tuning With the baseline controller structure available, the next step is to tune the free parameters of the individual control loops. During this process, an individual optimization problem is set up for the tuning of each control loop. This results in six optimization problems to be solved, as summarized in Table 11. Note that the proportional damping augmentations in roll and pitch are not tuned separately but included in the optimization problems of the corresponding tracking loops. For the inner loops a phase margin of at least 45° is demanded. As short period damping is relevant, a minimum of 0.6 is set as optimization constraint. For the roll motion a fast response time of 1 s with good tracking capabilities (steady state error of 0.1) is defined. For the coordinated turn capabilities via the side slip angle feedback a single constraint on the disturbance rejection gain is applied. For the outer loops an adequate frequency separation commonly used in a cascade controller design is applied. The outer loops for controlling attitude and course angle are designed to be five times slower than the inner loops, leading to a corresponding bandwidth or response time constraint. Finally, the auto-throttle is a little more involved due to the complex engine dynamics. Therefore, a model matching problem using the non-linear simulator is used which aims to minimize the recorded error between the desired and achieved response in the simulation.

Table 11: Overview of the six defined optimization problems with the number of free parameters and optimization criteria within the model based design procedure of the baseline controller.

Channel	Structure	Free Parameters	Criteria
Pitch Attitude Control incl. Pitch Damping	PI P	8	Damping ration of 0.6 Phase margin of 45°
Roll Attitude Control incl. Roll Damping	P P	4	Response time of 1s, steady state Error of 0.1, phase margin of 45°
Yaw Control	PID	9	Disturbance rejection gain
Auto-Throttle	2 DOF-PID	5	Model matching error
Altitude	PI	6	Bandwidth criterion
Course Angle	PID	9	Response time of 5 s

Baseline control flight test results The baseline controller has been tested in an intensive flight test campaign, where the separate loops have been sequentially engaged in the different flights. It is important to emphasize, that the controllers have been designed and tuned based on the available mathematical models of the aircraft, which is only an approximation of the real dynamics. Therefore, several adjustments (fine-tuning) of the control gains might be necessary in order to improve the performance of the baseline controller. This can be performed smoothly due to the hierarchical structure of the controller. Accordingly, different tuning were tested and compared to each other for the best achievable design.

Augmented Mode Flight Tests The first step of the testing is the so called stability augmented flights, where three inner-loops are engaged (i.e. lateral, longitudinal and yaw) and the pilot directly controls the pitch and roll behaviour of the aircraft, instead of the control surfaces.

In Flight Test 11 (FT11) three different lateral inner loop controllers were tested:

- The gains of AP1.1. were tuned down, for a slower response, in order to safely check the functionalities.
- AP1.2. gains are higher than AP1.1. and provides a more aggressive (hence less robust) tracking performance.
- AP1.3. uses a different Look-Up-Table mapping of the baseline control signals onto the surface deflections.

The three controller is compared in Figure 98, where the roll angle tracking and the corresponding aileron deflections are shown for each configuration. Based on the flights the pilot and the flight test crew have agreed that the behaviour of AP1.3 was acceptable, therefore this version was used in the later flights.

The post-flight numerical analysis supported the findings of the flight test crew, as AP1.2. and AP1.3. provided almost the same bank angle performance with Root Mean Square (RMS) error of 4.15 and 5.6 degrees, respectively, while AP1.1.'s error was approximately 7.3 degrees.

Due to the negligible cross-coupling between the lateral and longitudinal axes, the inner and outer-loop control law for the longitudinal motion could be tested independently. That being said, AP2 of FT11 consisted the pitch attitude inner loop and the engagement of the altitude hold outer loop. The altitude hold feature was tested for the first time, where the aim of the control was to hold the GPS altitude

registered at the moment of the baseline autopilot's engagement. The corresponding flight data is given in Figure 99.

As it can be depicted the altitude hold function was working properly, holding the constant altitude with an RMS value of 6.6 meters, due to the various gust disturbances. The performance of the pitch attitude control was very promising as well, with mean-error value of -0.016 degrees. Note that the blue reference signal of the θ tracking plot in Figure 99 is provided by the outer-loop controller in order to maintain the desired altitude. The lower subfigure also shows the computed elevator deflections of the aircraft, where no saturation was observed.

However, the flight test also pointed out that the outer loop's bumpless transfer mechanism was not properly implemented. When the pilot engaged AP2, the aircraft nosed down slightly, resulting in a sharp transient as seen in Figure 99. This feature has been corrected.

The third function of the baseline controller, which was tested in Flight Test 11 was the sideslip loop. This loop was engaged during the testing of AP1.1, AP1.2, AP1.3 and AP2 as well: Figure 100 shows the recorded data. The goal of the controller is to maintain 0 sideslip angle, which is clearly achieved: the mean error of β was approximately 0.07 degrees when the baseline controller was turned on, compared to the 1.8 degree of uncontrolled value.

Altitude Tracking and Autothrottle Tests The goal of Flight Test 12 was to check further functionalities of the baseline control structure. First, the reference tracking properties of the altitude hold control loop was tested. During this test, the reference altitude was changed by ± 25 meters, instead of the constant value applied in FT11. Figure 337 shows the tracking performance, where an approximate 10 seconds of settling time was observed with a permanent error of ≈ 3 meters. This latter result implies a further fine tuning of the integral part of the longitudinal outer loop. Besides this phenomena, the altitude loop can be considered functional.

Testing of the autothrottle loop involved three different control laws:

- A 2-degrees-of-freedom PID controller with low gains was tested first in order to first check the basic response of the control structure. This version is considered as a robust solution.
- A 2-degrees-of-freedom PID controller with higher gains is labelled as performance solution, where the response is more aggressive.
- Lastly a Total Energy Control solution was also implemented, where the altitude hold and the speed control are coupled together.

Figure 102 compares the control performance of the three controllers. The results are matching the model based expectations clearly: the 'robust' solution provided a slow tracking response for the commanded $4\frac{m}{s}$ step change in the airspeed, compared to the 'performance' version of the same control structure. Based on the post-flight numerical analysis, the TECS solution performed the best: the mean speed error was $-0.55\frac{m}{s}$ only, smaller than the robust ($-1.5\frac{m}{s}$) and the performance ($-1\frac{m}{s}$) errors.

However, the flight tests revealed a few shortcomings of the autothrottle loop:

- The saturation limits of the 2-DOF-PID solutions were implemented wrongly, hence these controllers could not use the entire RPM range of the engine. The problem has been fixed and further flight tests will be performed in order to evaluate these controllers.
- The engine was spooling down when the autothrottle loop was engaged (see the drops of ECU RPM values in Figure 102. The cause of this phenomena is again the bumpless transfer mech-

anism. In the new version of the autothrottle controller, an integrator tracking solution is implemented in order to avoid sudden drops in the RPM.

- The control action of the autothrottle (for each configurations) showed large variation in the RPM values. Figure 103 illustrates the problem. It can be clearly seen that the requested and the commanded RPM values are moving with a different phase, this indicates that the estimated delay for the engine dynamics differs from the actual value. To overcome this problem an updated engine model is needed and accordingly the re-design of the autothrottle loop.

Based on the experiences of the pilot and the flight test crew, the 'performance' controller has been selected for future future flight tests on the site.

Course Angle Flight Test Flight Test 14 was dedicated to test the course angle hold and tracking capabilities of the baseline controller. This has been performed in two consecutive steps.

First, the course angle loop was tested through a reference step change, and a coordinated turn maneuver. Figure 342 shows the inner and outer loop performance of the lateral loops. As it can be seen, first the heading reference angle χ was changed in a step-like fashion, which was tracked by the controller properly. After this successful test, a coordinated turn maneuver was performed, where the course angle was changed incrementally to fly a complete circle with the T-Flex aircraft. It can be depicted in Figure 342 that the inner loop's tracking performance is excellent, while the outer loop follows the reference signal with a slight delay. Figure 105 shows the trajectory of the aircraft during the maneuver.

Upon the successful testing of the course angle loop, the navigation logic was tested. The goal was to fly the complete horserace pattern in a fully automated manner, i.e. all baseline loops engaged. Figure 343 shows the flight trajectory in North-East coordinate system, where one can clearly observe that the entire functionality of the baseline works smoothly and is able to fly the desired pattern fully autonomously.

Preparation for Flutter tests As the baseline functionalities have been tested with acceptable control performances, further a flight test was performed in order to facilitate future flight tests for flutter control. The objective of the test was to gradually increase the speed of the aircraft and see the behaviour of the baseline controller. This test is useful in the preparation of the flutter flight tests, where the speed will be increased in the straight legs of the horserace pattern. In addition, these tests are also validating the range of the baseline controller. As mentioned previously, the gains of the controller are scheduled with the indicated airspeed, hence expanding the speed range expands also the domain of the baseline controller.

In order to keep the aircraft within the view sight of the pilot, full circles have been flown with increasing speed. figures/D106 107-109. Figure 107 shows the lateral inner-outer loop performances during the flight, where similar performance has been observed as in the previous flight tests. The corresponding trajectory of the aircarft is shown in Figure 108 with the multiple full circles. Lastly, Figure 109 shows the speed profile during the flight. It can be seen that the autothrottle was able to track the increasing speed reference until $50\frac{m}{s}$, but the control performance was diminished above this speed. This is due to the previously noticed incorrect saturation limit on the RPM values. Nevertheless, it is also worth to mention that the tracking performance below $50\frac{m}{s}$ was better than in earlier flight tests, this is due to the design process of the autothrottle, which was tuned through non-linear optimization and with all other baseline loops engaged.

Validation of data driven wingshape estimation by analytic models One of the main goals of the FLiPASED project is to develop drag reduction control laws for aircraft with highly flexible wings. The main motivation of this chapter is to present two different approaches, one relying on purely theoretical models, the other using experimental data utilizing machine learning tools, to estimate the flexible dynamics of the T-Flex demonstrator which can be then used for the design of a wing shape controller for drag reduction purposes.

The most straightforward solution for designing a state predictor is the Kalman filter ([59]) for linear systems and the extended Kalman filter (EKF) in the case of nonlinear systems, which is widely used for inertial estimation of wing shape ([64]). However, it has two main drawbacks. First, it requires the exact mathematical state-space description of the nonlinear system, which might not be available or simply its use is computationally too expensive. The second drawback is that the EKF requires knowledge of noises and disturbances related to observations and states. To solve the first issue, the approximation of the full, nonlinear system with a Linear Parameter Varying (LPV) model ([106]) can be considered, as it can significantly ease the computational burden, while being suitable for use with an EKF ([96]). However, noise information is still required.

Data-driven approaches have the great advantage that they can be used for inertial odometry ([23]) and inertial aided navigation ([124]) problems without needing any specific information about model or observation uncertainties. To utilize this advantage the new KalmanNet architecture was created by [90], which is based on Kalman filtering, but it uses a Recurrent Neural Network (RNN) to estimate the Kalman gain. As a result, it does not need any information about the noises and model uncertainties present. The standard KalmanNet architecture uses linear layers and a Gated Recurrent Unit (GRU) to be able to establish correlations between data samples in time. The main novelty of this research are the followings. First, we apply the KalmanNet architecture to a complex LPV model of a real-life system with high dimensionality. This required a new loss calculation to ensure the stability of the whole state predictor, the slight modification of the layer dimensions in order to decrease computational burden and the implementation of a hyperparameter optimizing algorithm as well. Second, we propose a different neural network architecture for the KalmanNet which uses 1D convoultional layers alongside the GRU layer, since 1D convolution can be effectively used for processing timeseries data ([98]).

This chapter presents the LPV-based EKF and the KalmanNet for predicting the modal coordinates and aerodynamic lag states of the nonlinear model of an Unmanned Aerial Vehicle (UAV) T-Flex, which was created within the FLEXOP project for demonstrator purposes. ([119]) The training of the neural network was done in Python with PyTorch, while testing was carried out with MATLAB, Simulink simulations. The chapter is organized as follows. In Section 2.1.3, the dynamic model of the FLEXOP demonstrator is presented. Section 2.1.3. introduces the reduced, LPV model of the original nonlinear system, and explains the idea of the LPV-based EKF. In Section 2.1.3. the KalmanNet's basic structure is summarized with the two different neural network architecture: a linear one and a convolutional one both utilizing a GRU cell. Section 2.1.3 presents the results of the modal coordinate and lag state estimations. The accuracy of the LPV-based EKF and the KalmanNet is compared. Conclusions are drawn in Section 2.1.3.

T-FLEX demonstrator dynamic model The chosen system for our research is the nonlinear, state space representation of the FLEXOP demonstrator aircraft. The model consists of the following main parts: states that are responsible for the description of the rigid body dynamics; states related to flexible dynamics and aerodynamics, and finally, states that represent the control surface inputs and their first derivatives. The state vector is denoted as $\underline{x} \in \Re^{48}$. The rigid body motion is represented with a 6-DOF model with 12 states: states of translational and angular velocities, position, and orientation.

The states which describe the flexible dynamics are the modal coordinates and their first derivatives. Due to the reduced order modelling only the six most significant modal coordinates and two aerody-

namic lag states were considered. This is denoted as $\underline{x}_{\text{flex}} \subset \underline{x}$ and $\underline{x}_{\text{flex}} \in \Re^{14}$. The main objective of this research is the estimation of $\underline{x}_{\text{flex}} = [U_{f1}, U_{f2}, U_{f3}, U_{f4}, U_{f5}, U_{f6}, \dot{U}_{f1}, \dot{U}_{f2}, \dot{U}_{f3}, \dot{U}_{f4}, \dot{U}_{f5}, \dot{U}_{f6}, \text{lag}_1, \text{lag}_2]^T$. Further details of the modeling and model order reduction are given in [119], [108] and [71].

The elements of the input vector, $\underline{u} \in \Re^{19}$, of the T-Flex aircraft model are (Figure 110): two landing gears (GearR/L), two landing gear wheelbrakes (WheelbrakeR/L), two airbrakes located on the aircraft's fuselage (AirbrakeR/L) and one turbofan engine (Throttle). The demonstrator has 12 control surfaces: four-four ailerons (AileronR/L) on each wing and on the V-tail two-two 'ruddervators' (TailR/L). From the inputs, the landing gear-related ones are insignificant in our research since the estimation of the structural dynamics is only conducted during airborne operations.

The output vector, $\underline{y} \in \Re^{64}$, of the demonstrator model is made up from the following elements. It has 23 rigid body related outputs, which provide information about the aircraft's position (x_E, y_E, z_E), orientation (ϕ, θ, ψ), translational (v_N, v_E, v_D) and angular velocity (p, q, r), and acceleration (a_{xB}, a_{yB}, a_{zB}). Furthermore, the course angle (χ), angle of attack (α), sideslip angle (β), air (p_a) and total pressure (p_T), barometric altitude (h_{baro}), indicated (v_{IAS}) and the true airspeeds (v_{TAS}) are measured as well. Each wing of the demonstrator has six-six inertial measurement units (IMUs). An IMU provides acceleration and angular velocity data around the x-, y- and z-axis of its coordinate system. We opted for such an IMU configuration, where the IMUs on the leading-edge measure accelerations in the x, y, and z directions, while the IMUs on the trailing-edge provide angular velocity data around the x- and y-axis, and acceleration data in the z direction. The exact location of the IMUs can be seen in Figure 110 as well. In addition, the wingtip coordinates can be measured with a mono camera for preventing acceleration-based estimation errors from diverging in time ([64]). On each wing, the coordinates of four wingtip points are measured in each direction.

Model based estimation of flexible dynamics

LPV model The linear parameter varying (LPV) model is an approximation to describe the behaviour of a nonlinear system ([106]). It is essentially a pointwise linearization of a state-space system: the nonlinear system is linearized at different trim points that are defined by - the so called – scheduling parameters. The scheduling parameters create a multidimensional grid, and a linear, state-space model is assigned to every grid point. The state-space description of a discrete time LPV system is described by

$$\begin{aligned} \underline{x}[k] &= A(\underline{\rho}[k])\underline{x}[k-1] + B(\underline{\rho}[k])\underline{u}[k], \\ \underline{y}[k] &= C(\underline{\rho}[k])\underline{x}[k] + D(\underline{\rho}[k])\underline{u}[k], \end{aligned} \tag{14}$$

where $\underline{\rho}[k]$ is the time varying vector of the scheduling parameters in time step k. $A \in \Re^{48\times48}$, $B \in \Re^{48\times19}$, $C \in \Re^{64\times64}$ and $D \in \Re^{64\times19}$ are the state space matrices of the LPV system that are dependent of the current $\underline{\rho}[k]$ vector. $\underline{x}[k]$ denotes the state vector, $\underline{u}[k]$ the system's input vector, while $\underline{y}[k]$ is the system's output vector in time step k.

In our work, we created an LPV approximation of the nonlinear bottom-up model of the T-Flex demonstrator aircraft with two scheduling parameters, $\underline{\rho} \in \Re^2$: the true airspeed ($v_{\text{TAS}}$) and the roll angle ($\phi$) sensor outputs. The grid for the LPV model consisted of airspeed values from 30m/s to 50m/s with a 1m/s resolution while the roll angles from 0° to 40° with 10° resolution. Then the nonlinear model was trimmed at each grid point. The resulting LPV model structure was then further refined to 0.1m/s and 1° resolution with the spline interpolation method of the LPVTools MATLAB toolbox ([4]).

LPV-based Kalman filtering For the model-based wing-shape estimation, an extended Kalman filter (EKF) was used. The EKF pipeline requires the full, nonlinear state-space description of the system

as well as information about the model noise and observation noise in the form of noise covariance matrices. The nonlinear system's state-space representation with time discretization in time step k:

$$\begin{aligned} \underline{x}[k] &= \mathbf{f}(\underline{x}[k-1], \underline{u}[k]) + \underline{w}[k] \\ \underline{y}[k] &= \mathbf{h}(\underline{x}[k], \underline{u}[k]) + \underline{v}[k]. \end{aligned} \tag{15}$$

The nonlinear function $\mathbf{f}(.)$ is called state-transition function, while $\mathbf{h}(.)$ is called state-observation function. The $\underline{w}[k] \in \Re^{48}$ and $\underline{v}[k] \in \Re^{64}$ vectors are the model noise and observation noise vectors respectively. However, the explicit mathematical description – the nonlinear state-transition and state-observation functions – of the T-Flex demonstrator was not available to us, therefore a unique approach was necessary for the design of the EKF.

The general framework of the EKF consists of two main steps: *prediction* and *update*. In these steps, pointwise linearization is used to approximate the behaviour of the nonlinear system. More precisely the Jacobians of the nonlinear state-transition and state-observation functions are calculated to get the linear, state-space matrices $A[k]$, $B[k]$, $C[k]$ and $D[k]$ at each time step. In the *prediction* step the prior state estimation is calculated using the inputs of the current time step and the estimations from the previous time step with

$$\underline{\hat{x}}[k|k-1] = \mathbf{f}(\underline{\hat{x}}[k-1|k-1], \underline{u}[k]). \tag{16}$$

The prior state estimation covariance $P \in \Re^{48\times 48}$ is

$$P[k|k-1] = A[k]P[k-1|k-1]A[k]^T + Q. \tag{17}$$

In the *update* step, first, the innovation

$$\underline{\tilde{y}}[k] = \underline{y}[k] - \mathbf{h}(\underline{\hat{x}}[k|k-1], \underline{u}[k]) \tag{18}$$

is calculated. then the near-optimal Kalman gain, $K_G \in \Re^{64\times 48}$

$$K_G[k] = P[k|k-1]C[k]^T(C[k]P[k|k-1]C[k]^T + R)^{-1}. \tag{19}$$

With the help of the Kalman gain, the posterior state vector

$$\underline{\hat{x}}[k|k] = \underline{\hat{x}}[k|k-1] + K_G[k]\underline{\tilde{y}}[k], \tag{20}$$

and state prediction covariance

$$P[k|k] = (I - K_G[k]C[k])P[k|k-1] \tag{21}$$

is computed. In the equations, the $Q \in \Re^{48\times 48}$ and $R \in \Re^{64\times 64}$ matrices are the model and the observation noise covariance matrices respectively.

To obtain an appropriate pointwise linearization we used our LPV model. During simulation, the true airspeed and roll angle is measured at each time step which then can be used to select an approximating linear system from the LPV model. The selected model's state-space matrices are fed to the EKF as the current $A[k]$, $B[k]$, $C[k]$ and $D[k]$ matrices. Then the EKF conducts the prediction and update steps. For acquiring the model noise matrix both the nonlinear and the LPV models were simulated with doublet inputs on the control surfaces and then the measured outputs and states were compared, and variances of the differences calculated. For the observation covariance matrix, the T-Flex's onboard sensors' noise variances were used. These were specified based on the sensors' datasheets. Note that we used the assumption that both noises have 0 mean, normal distributions, and the noise vectors at each time step are mutually independent.

KalmanNet architecture The other approach for estimating the flexible dynamics of the demonstrator is to use artificial intelligence, more precisely a neural network. Our choice was to use the relatively new KalmanNet architecture ([90]). KalmanNet combines Kalman filtering with a neural network as it still uses the current inputs and observations for giving state estimations, however, the near optimal Kalman gain is provided by a trained recurrent neural network (RNN). The main advantage of this is that KalmanNet does not require either the model (Q) or the observation noise covariance matrices (R) and it can effectively overcome any uncertainties or errors in the model of the dynamic system while retaining engineering insight about the physical system.

The KalmanNet pipeline is the following. It still consists of a *prediction* and an *update* step just like a Kalman filter. In the prediction step however only the prior state prediction (16) is calculated, the state prediction covariance (P) is not. In the update step, first the innovation difference ($\Delta\underline{y}[k] \in \Re^{64}$) and the forward update difference ($\Delta\underline{x}[k] \in \Re^{48}$) are computed:

$$\Delta\underline{y}[k] = \underline{y}[k] - \underline{\hat{y}}[k|k-1] \tag{22}$$

$$\Delta\underline{\hat{x}}[k] = \underline{\hat{x}}[k-1|k-1] - \underline{\hat{x}}[k-1|k-2]. \tag{23}$$

These act as the input features for the recurrent neural network. Furthermore, the roll angle (ϕ) scheduling parameter was also used as an input feature in order to make the handling of turning manoeuvres easier. The RNN then provides the Kalman gain in each time step. With the Kalman gain and using the innovation, the *a posteriori* state prediction vector is calculated as in (18) and (20) respectively. As it can be seen, neither the state estimation covariance matrix (P) nor the noise covariance matrices are required: the whole pipeline works without any information about the model or observation noises.

The standard Kalman gain predicting neural network presented in [90] uses a Gated Recurrent Unit (GRU) as the recurrent layer and linear layers with Rectified Linear Units (ReLU) as activation function. The architecture is the following: it has a linear layer as the input layer with ReLU activation followed by the GRU. After the GRU layer, there is another linear layer with ReLU activation, then the linear, output layer. We slightly decreased the dimensions of each layer compared to the proposed architecture. The reason behind this modification was mainly memory consumption related. Since our aircraft model has a relatively high number of states (48) and outputs (64) using the original layer dimensions, we frequently ran out of GPU memory, while training with CPU was extremely slow.

Apart from the original *linear* architecture, we implemented a different neural network loosely based on the one presented in [124]. The network architecture still uses a GRU cell, however instead of linear layers it uses 3 convolutional blocks at the beginning. A convolutional block consists of a 1D convolutional layer followed by a ReLU activation function. After the ReLU a Batch Normalization layer is used which is followed by a Dropout layer with 0.25 dropout probability. A fully connected layer is only kept at the end of the network for providing the Kalman gain matrix. The kernel size for each 1D convolution layer is seven. As the 1D convolutional layer requires a trajectory, or time-window of input features, simply using the previously mentioned forward update difference ($\Delta\underline{y}[k]$), innovation difference ($\Delta\underline{\hat{x}}[k]$) and roll angle ($\phi$) input features of the current timestep is not adequate. Therefore, we used the input features of the current timesteps and the input features form the previous 19 timesteps in the time-window buffer.

The full KalmanNet pipeline with the convolutional neural network is presented in Figure 198. The number of features is shown below the convolutional blocks, the pool size below the max pooling layer. The number of units is indicated underneath the GRU and the linear layer. The dropout rate is shown below the dropout layer.

Training data Training, validation, and testing datasets were generated using the T-Flex's high-fidelity nonlinear Simulink model. In order to create a rich dataset, while having realistic flight conditions, an '8-shaped' track was generated using the baseline controller of the aircraft also implemented in Simulink.

To make each dataset different, randomized windgust and turbulence disturbances were used as well, together with Gaussian sensor noise, based on the flight test results of the demonstrator [6]. The main purpose of applying wind loads is having disturbances that cannot be incorporated into any covariance matrix, and generating trajectories that are more realistic. Going through the generated track takes the aircraft roughly 120s with the initial velocity of 42m/s, so the duration of the simulation was set accordingly. The sampling time was set to 5ms, which results in a 24000-sample long trajectory for each dataset. This was then split into 20, 1200-sample (6-second) long batches. For training, in each epoch 8 batches were randomly selected from the total 20. However, validation, as well as testing, was conducted on the whole 20-piece, 6-second-long trajectory in order to get meaningful information about the architecture's performance.

Training details For the two neural network architectures the training parameters were set with the use of a custom made hyperparameter optimization algorithm based on *RayTune*. The hyperparameter optimization had 20 runs, each lasting for 25 epochs. Otherwise, the hyperparameter optimization used the same pipeline as normal training runs.

The optimizer algorithm was ADAM for both architectures. In order to avoid overfitting, weight decay was used. The prediction accuracy was calculated with mean squared error (MSE) function. However – although the linearized aircraft model is a stable system, the system's poles are relatively close to the unstable region. So, a stability criterion was added to the MSE loss function. It is possible to describe the complex system of the aircraft model joined with the Kalman filter with an error system:

$$\underline{e}[k+1] = (A - K_G C)\underline{e}[k], \tag{24}$$

where K_G is the Kalman gain, $\underline{e}[k] \in \Re^{48}$ is the state prediction difference at time step k. If the error system's state transition matrix $(A - K_G C)$ has any unstable poles, then the whole system is unstable. Hence, the MSE loss was extended with the distance of the error system poles from the boundary of stability if it is larger than 0, thus making the loss value larger if the computed Kalman gain results in an unstable error system. This is especially useful for the convergence of the training.

The error metrics were defined in decibels for the sake of convenience during plotting because the freshly initialized network tends to produce large errors. It is simply calculated with the following formula:

$$\text{loss}_{\text{MSE}}^{\text{dB}} = 10\ \log_{10}(\text{loss}_{\text{MSE}}). \tag{25}$$

Of course, the metric was solely used for evaluation and plotting. For optimizing the network weights, the standard MSE loss value was used during backpropagation.

For initializing each layer's weights, standard normal distribution was used. However, as the whole architecture incorporates a discrete time system, it was very sensitive to the initial weight values. Therefore, standard deviation of the normal distribution for the initialization had to be chosen to be very small $(5 \cdot 10^{-6})$ for avoiding the otherwise highly diverging training process.

It is important to mention that the 1st architecture's performance proved to be more stable than the 2nd which had the tendency to get stuck in local optima. So, to overcome this issue the reduction of the learning rate during training was necessary in the case of the 2nd network. The threshold was set at -42dB – according to the decibel-based error metric – and the reduction factor was 0.05. The new learning rate was calculated as $\text{lr}^{\text{new}} = \text{factor} \cdot \text{lr}^{\text{old}}$. The used hyperparameters for each neural network architecture are presented in Table 12 .

Table 12: Hyperparameters

Hyperparameter	Linear	Convolutional
Learning rate	$3.2 \cdot 10^{-6}$	$1.5 \cdot 10^{-7}$
Weight decay	$7.5 \cdot 10^{-5}$	$9.5 \cdot 10^{-5}$

Results

LPV-based EKF The LPV-based EKF's performance was evaluated on the test dataset generated for the KalmanNet. This means that during the simulation, the aircraft followed the same 8-shaped track with wind and turbulence disturbances independent from the training data. The initial conditions were 42m/s flight speed at 800m altitude, with 2° course angle. The whole simulation lasted for 120 seconds which corresponds to 1 full lap around the track. The results of the EKF-based state predictions are shown in Figure 112, where the data with the 'nonlinear' label show the states of the nonlinear model, while the 'EKF' show the states estimated by the filter. Since the main purpose of the observer design is to observe the flexible dynamics of the states, only the results for these states are presented. The first 4 modal coordinates are plotted where U_{f1} is the 1st symmetric bending and U_{f2} the 1st asymmetric bending mode. U_{f3} denotes the 1st symmetric torsion mode and U_{f4} is the 1st asymmetric torsion mode. The 2 aerodynamic lag states are plotted as well.

From the results, it can be concluded that the designed filter accurately predicts the modal coordinates and the aerodynamic lag states even in the presence of wind disturbances which's effects cannot be incorporated into the observation noise matrix. The predictions' root mean squared error (RMSE) for the plotted states is $7.62 \cdot 10^{-4}$. Minor errors occur only during turning manoeuvres in state lag_1. The reason behind these is that the LPV model is still just an approximation of the real, nonlinear system. However, these inaccuracies are inside the error tolerance for this problem.

KalmanNet For training, the generated 20 batches of 1200 sample long trajectories were used with a sampling time of 5ms. The inputs for the KalmanNet architecture were the observations and control surface and throttle inputs of the nonlinear model. The target for the network were the nonlinear model's states. During training, validation and testing the KalmanNet used the LPV model of the nonlinear model. For evaluation the initial trim condition of 42m/s true airspeed was used at 800m altitude with initial course angle of 2° just like in the case of the EKF.

Neural network with linear layers First, we tried the slightly modified original KalmanNet architecture which uses linear layers with the GRU. The training lasted for 200 epochs. Using an Nvidia Tesla V100 GPU with 32GBs of RAM, the whole procedure took 23 hours. The summary of the training is presented in Figure 113 (*left*). The previously discussed decibel-based metric was used for plotting.

The trained model was evaluated on the same dataset as the LPV-based EKF. The results for the first 4 modal coordinates and the 2 aerodynamic lag states can be seen in Figure 114. From the results, it can be seen that the neural network managed to produce comparable results with the EKF in the case of U_{f1}, U_{f3}, U_{f4} and lag_2. In the prediction of U_{f2} and lag_2 however a larger error is present between time step $8000 - 12000$. As, in this interval, the aircraft conducts a heavier acceleration, it is possible that the training data is not comprehensive and balanced enough to make the neural network capable of learning such manoeuvre. The RMSE value of the predictions is $1.67 \cdot 10^{-3}$.

Neural network with convolutional layers Second, we implemented the proposed network architectures with convolutional layers. In this case, the training duration was 100 epochs. That took 15 hours to complete using the V100 GPU. The 1D convolution expects a time series as an input, a 20-sample long time window was used which equals to 0.1s trajectory. Unfortunately, we could not use a larger window size, because we ran out of GPU memory (and training with CPU was not feasible, due to its extremely slow execution speed). The training graph is shown in Figure 113 (*right*) with loss values in decibels.

Testing was done with the same dataset as in the previous approaches. The results for the first 4 modal coordinates and the 2 aerodynamic lag states are presented in Figure 115. The results indicate the following: the network manages to give similar predictions in the case of U_{f1}, U_{f3}, U_{f4} as the LPV-based filter, but performs worse on U_{f3}, lag_1 and even lag_2. However, the prediction error of lag_2 in the $8000 - 12000$ interval is smaller than in the case of the linear architecture. The RMSE in this case was $1.69 \cdot 10^{-3}$ for the first four modal coordinates and the two lag states, which is somewhat bigger than the linear architecture's, but not significantly.

Data Driven vs. Model Based Estimation Conclusion To summarize, in this work we propose a model-based and a data-driven approach to estimate the flexible dynamics of a UAV with large wingspan and highly flexible wings. The model-based approach uses an LPV-based EKF while the data-driven solution utilizes the KalmanNet architecture with 2 different neural network setups. We showed that the EKF-based estimator is able to predict the flexible and aerodynamic lag states. The neural network-based approach is also capable of estimating the above-mentioned states, however, in the case of the U_{f3}, lag_1 and lag_2 larger errors are present. Comparing the two neural network it can be concluded that both provide relatively similar accuarcy. However, although the training of the *linear* network proved to be more stable, it requires almost double the training time as the *convolutional* one. It is important to mention, that the revision of the training datasets, and - generally - further research and datasets are needed to aquire better and more accurate results. Our long-term goal is to test both architectures in real-life flight data and then incorporate them in the T-Flex's FCC for real-time, airborne operations. With this, it will be possible to design a wing shape controller to minimize aerodynamic drag during flights.

Standardization Recommendations 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.

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.

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 116 shows the generated CPACS dataset.

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 [56] 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 117.

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.

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 118.

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 119 illustrates the connection between the mentioned tools.

Table 13: Tools, inputs and outputs in the aircraft model generation section

Tools	Inputs	Outputs
CATIA update	• CPACS dataset	• Folder containing updated CATIA model
FE model generation	• CPACS dataset • Folder containing updated CATIA model	• Folder containing containing bulk data files corresponding to the structure of wing
Panel generation	• CPACS dataset	• Folder containing containing bulk data files corresponding to the DLM of wing
Directory merge	• Folder containing containing bulk data files corresponding to the structure of wing • Folder containing containing bulk data files corresponding to the DLM of wing	• Folder containing containing bulk data files corresponding to the wing

In terms of the data-flow to each tool, the following Table 13 summarize the inputs and outputs to and from each of the tools.

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 14 and Figures 120-121 summarize the inputs and outputs to and from each of the tools.

Table 14: Tools, inputs and outputs in the NASTRAN aeroelastic model integration block

Tools	Inputs	Outputs
NASTRAN aeroelastic model integration	• CPACS dataset • Folder containing bulk data files corresponding to wing	• Folder containing model data - system matrices (stiffness, mass), aerodynamic panel model definition, outputs defining the condensation grids (summarized in Figure 121) • CPACS dataset
DLR-AE post-processing		• NASTRAN solution decks for modal, aeroelastic trim and flutter analyses

The ouputs from NASTRAN are contolled 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.

ASE model integration The ASE model integration section shown in Figure 122 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

Table 15 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 15: Tools, inputs and outputs in the ASE model integration block

Tools	Inputs	Outputs
Varloads model generation	• Folder containing model data - system matrices (stiffness, mass), aerodynamic panel model definition, outputs defining the condensation grids (summarized in Figure 121) • CPACS dataset	• 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	• 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	• directory with trimming results and corresponding linearized models
flutter analysis	• directory with trimming results and corresponding linearized models	• directory with flutter analysis results

Table 16: Tools, inputs and outputs in the flutter control design block

Tools	Inputs	Outputs
Modeling	• Folder containing LTI ASE model • CPACS dataset	• Folder containing the reduced order LTI ASE model • CPACS dataset
Flutter control design	• Folder containing the reduced order LTI ASE model • CPACS dataset	• directory with the resulting flutter suppression controller given in LTI structure • CPACS dataset
Flutter controller analysis	• Folder containing the reduced order LTI ASE model • directory with the resulting flutter suppression controller given in LTI structure • CPACS dataset	• directory with flutter analysis results • CPACS dataset

Flutter controller design The model order reduction and flutter control design blocks shown in Figure 123 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.

Table 16 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.

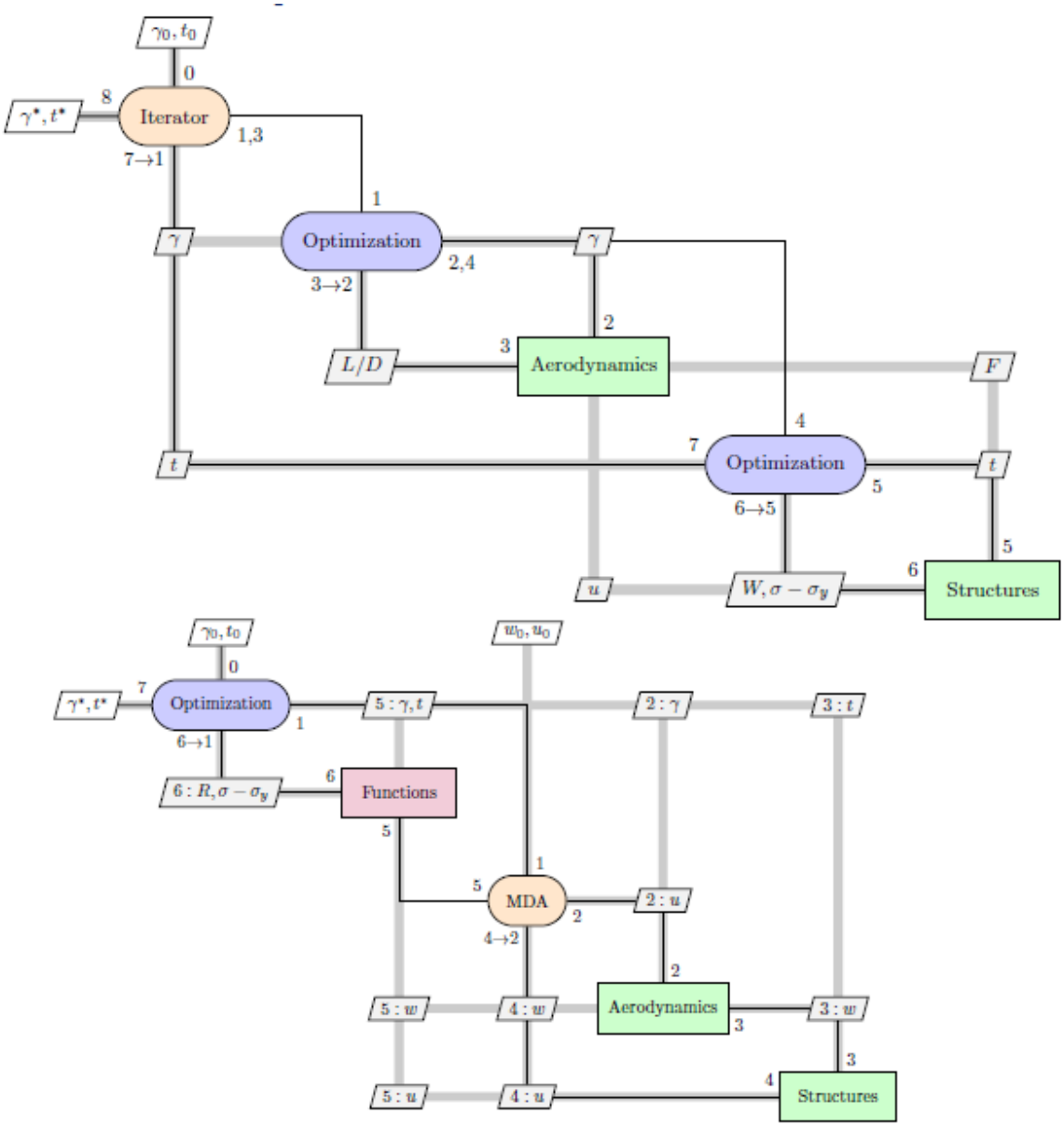


Figure 50: Sequential aero-elastic optimization vs. MDO framework [14]

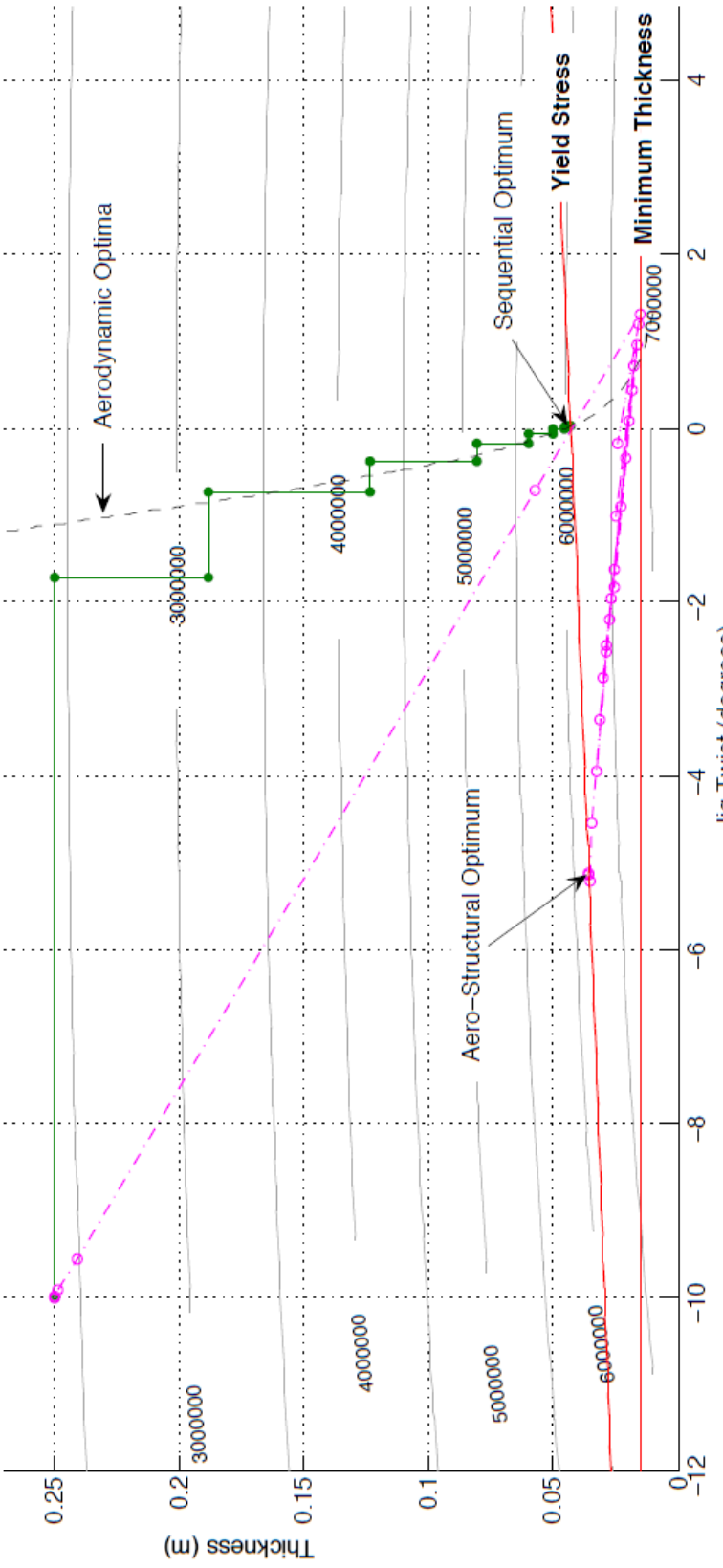


Figure 51: Iterations of sequential vs. global aero-elastic optimization (MDO) [14]

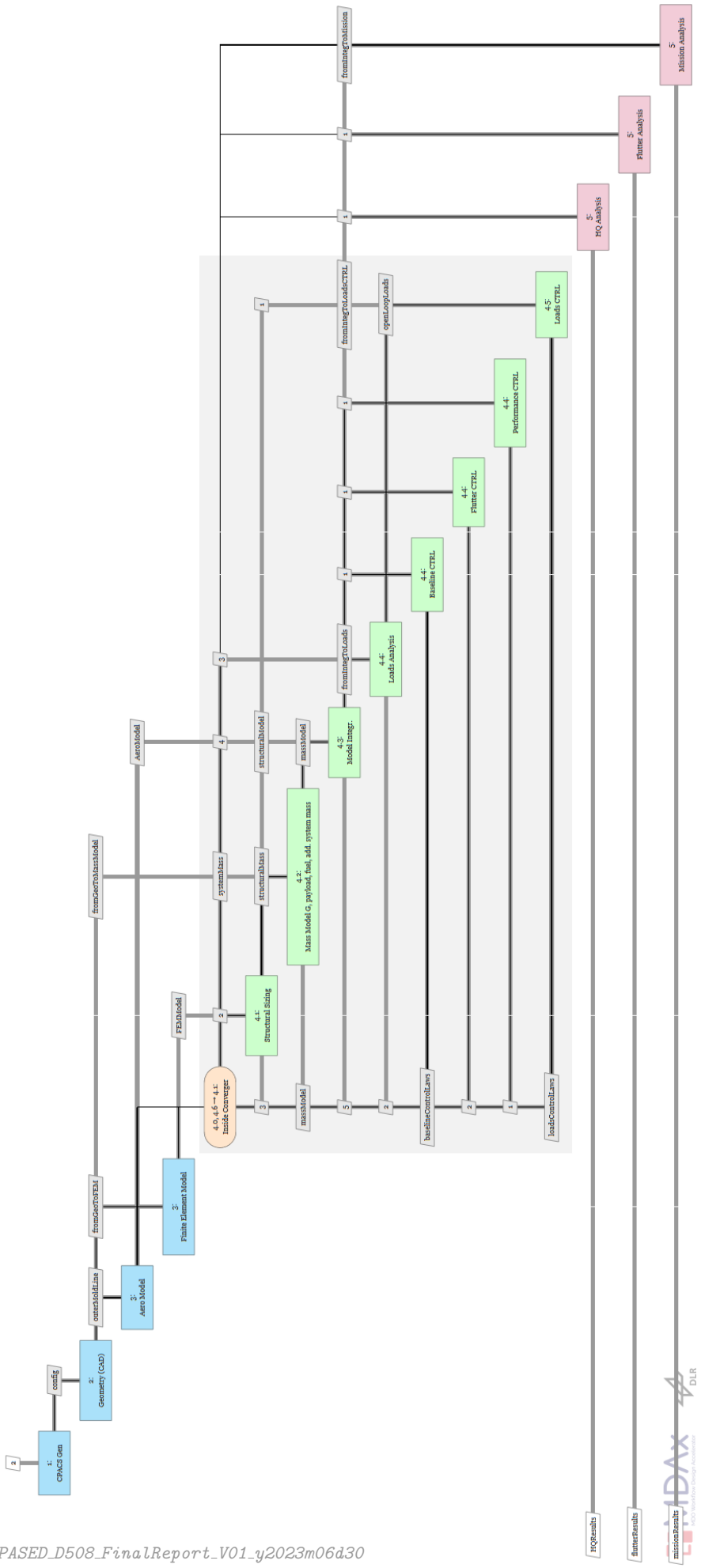


Figure 52: XDSM of scale-up workflow

Figure 53: Airbus XRF1 FEM model

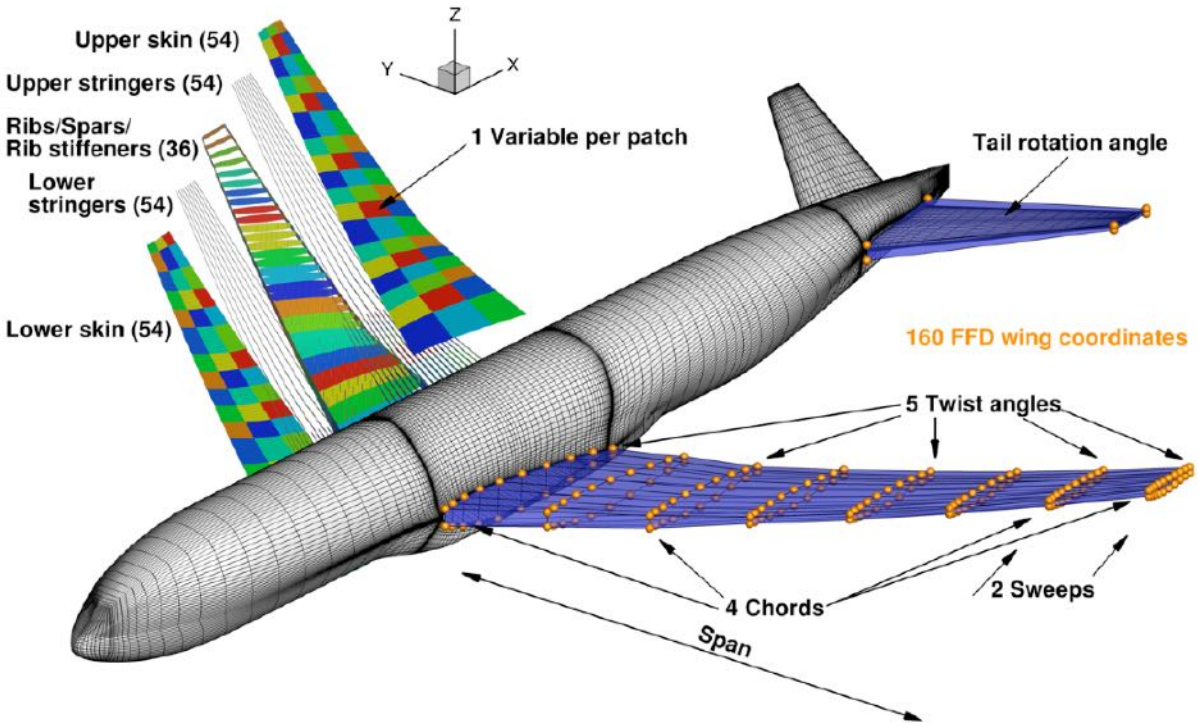


Figure 54: The aerodynamic and structural sizing design variables of the CRM MDO model [68]

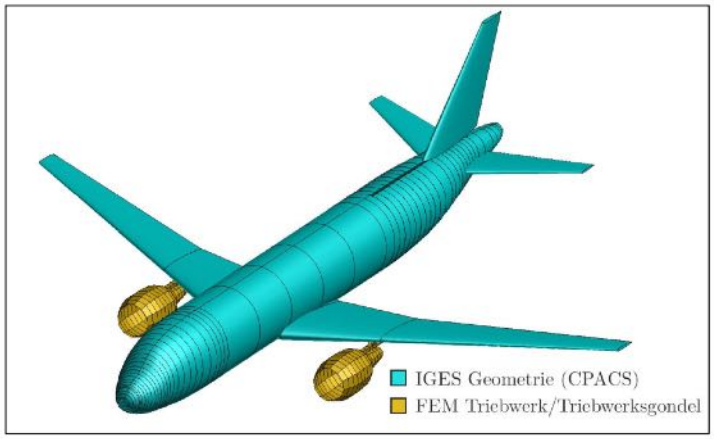


Figure 55: IGES-geometry of the D150-configuration

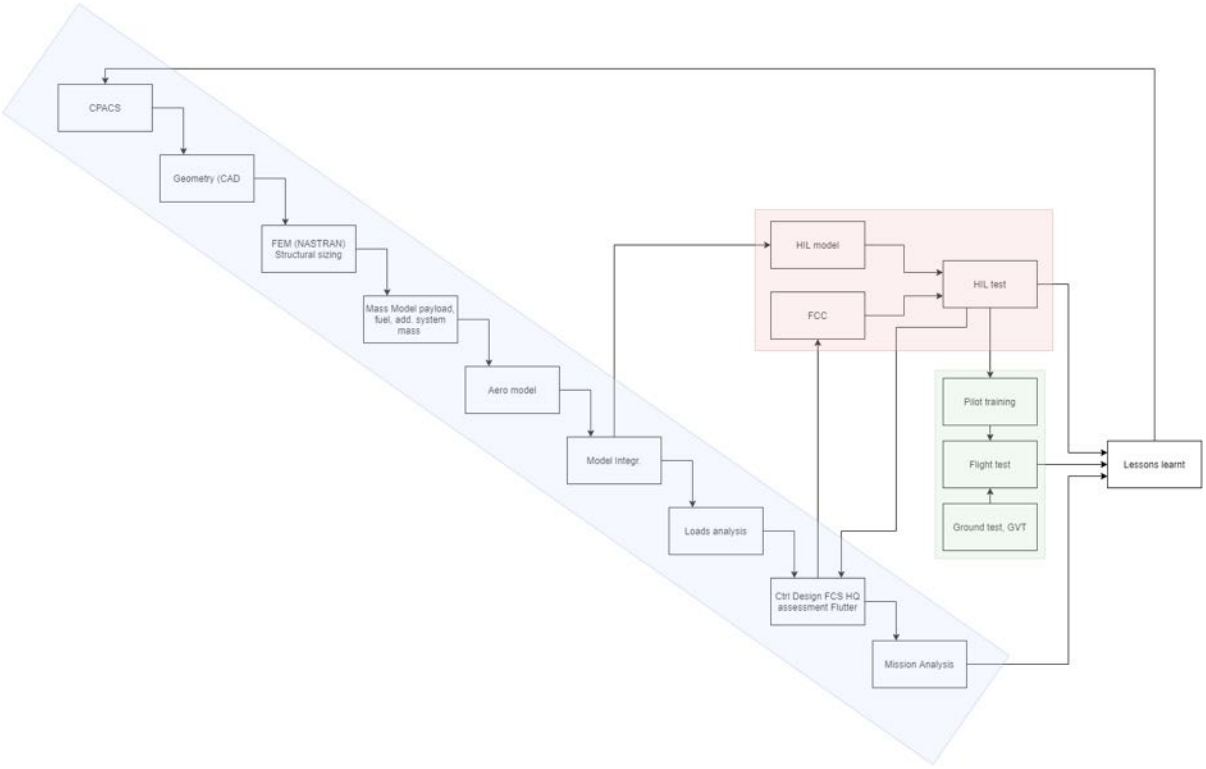


Figure 56: Workflow and inter dependency of tasks within FLiPASED (blue: MDO toolchain, red: simulation based testing and evaluation, green: physical testing)

Figure 57: Flight control surfaces of the Airbus A320 family, as well as the D150

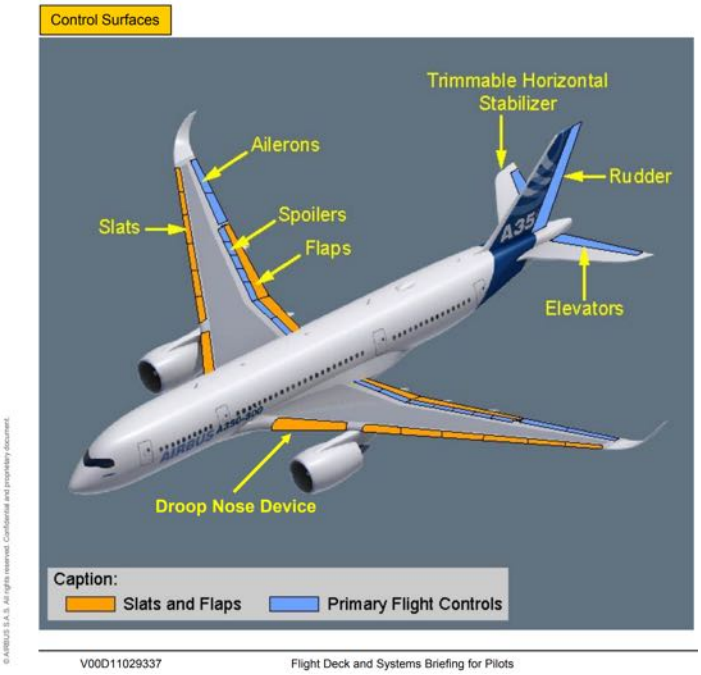


Figure 58: Flight control surfaces of a state-of-art widebody aircraft (Airbus A350)

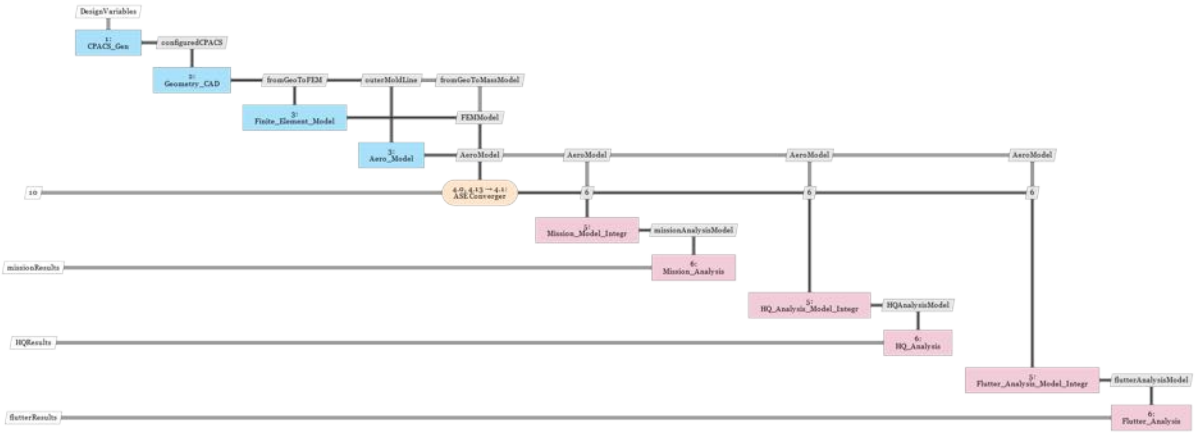


Figure 59: MDO Toolchain for demonstrator T-FLEX

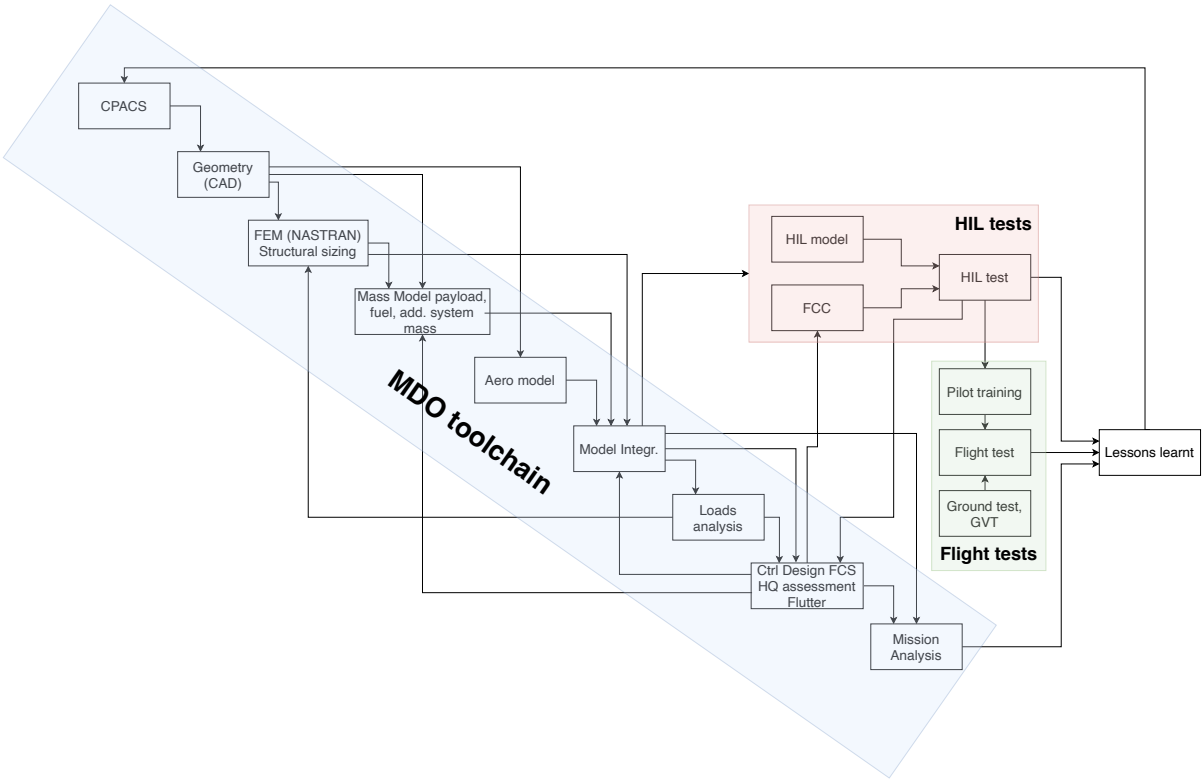


Figure 60: Toolchains developed in FLiPASED

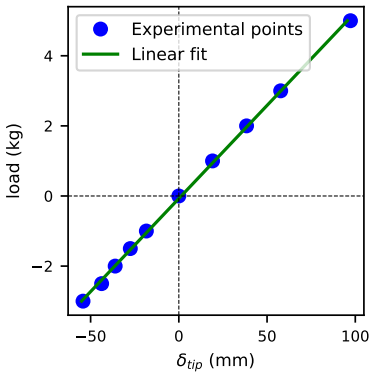


Figure 61: Displacement vs load at tip of the wing from static tests (-0 wing)

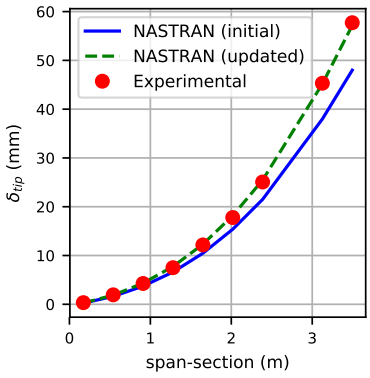


Figure 62: Span-wise displacement of wing under tip load for the updated model (-0 wing)

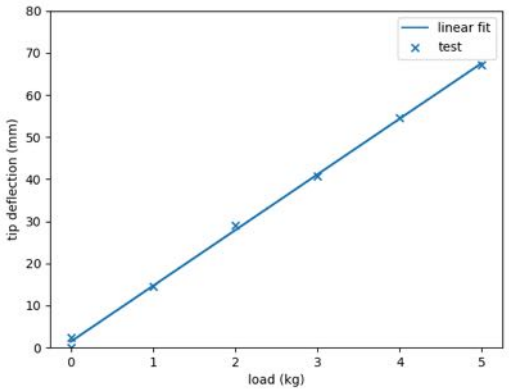


Figure 64: Displacement vs load at tip of the wing from static tests (-1 wing)

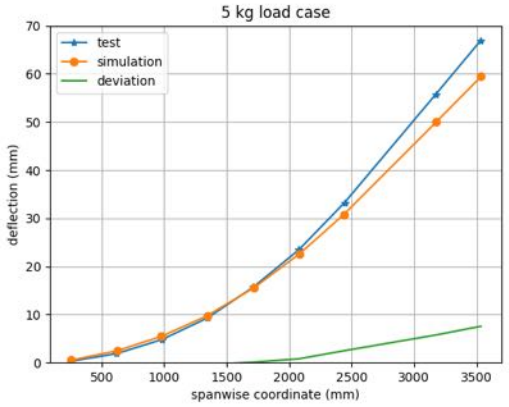


Figure 65: Span-wise displacement of wing under tip load (-1 wing)

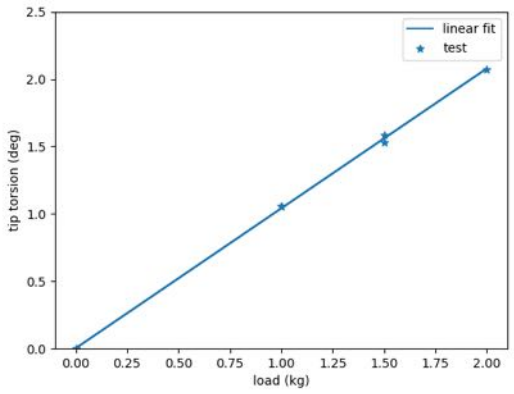


Figure 66: Torsion vs load at tip of the wing from static tests (-1 wing)

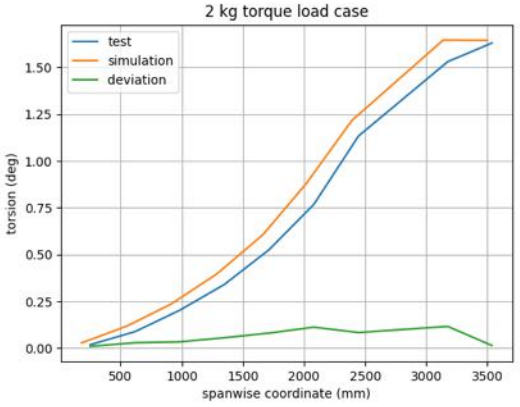


Figure 67: Span-wise torsion of wing under tip load (-1 wing)

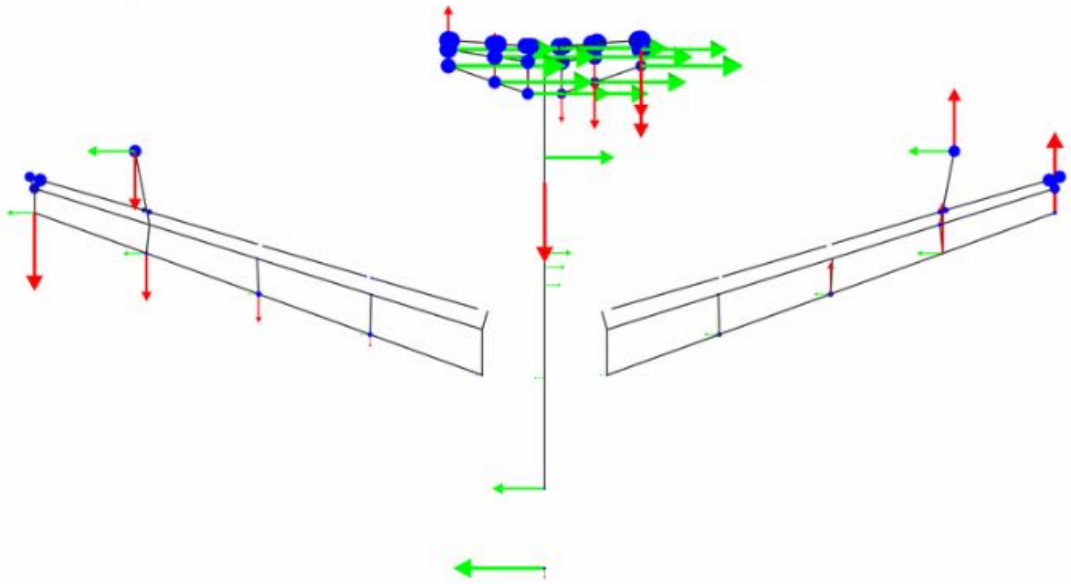


Figure 68: -1 aircraft 1n_wing_in-plane mode

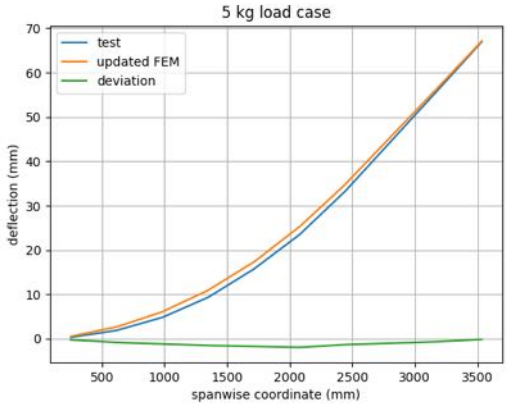


Figure 69: Span-wise displacement of wing under tip load (-1 wing)

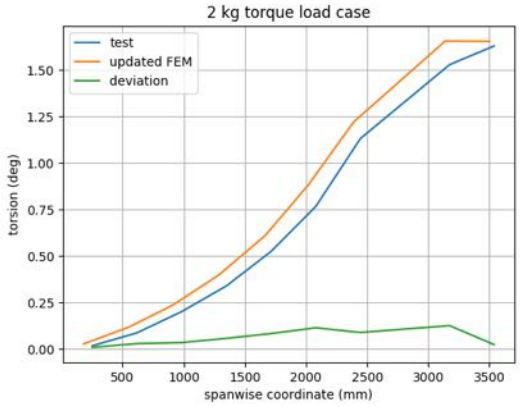


Figure 70: Span-wise torsion of wing under tip load (-1 wing)

Nr	Mode	GVT	FEM	deviation %	updated FEM	deviation updated %
0	2n wing bending-s	2.94	2.91	-1.19	2.74	-6.91
1	3n wing bending-a	7.57	8.15	7.7	7.66	1.13
2	wing torsion-s	10.27	10.5	2.2	10.43	1.6
3	wing torsion-a	10.73	10.61	-1.15	10.53	-1.86
4	4n wing bending-s	12.13	12.11	-0.19	11.42	-5.83
5	2n wing inplane-s	15.07	15.06	-0.05	14.32	-4.96

Figure 71: Comparison of eigen frequencies of the flexible modes: GVT vs FE model vs updated FE model of the -1 aircraft (*i*n - *i* nodes in the mode, s - symmetric, a - antisymmetric)

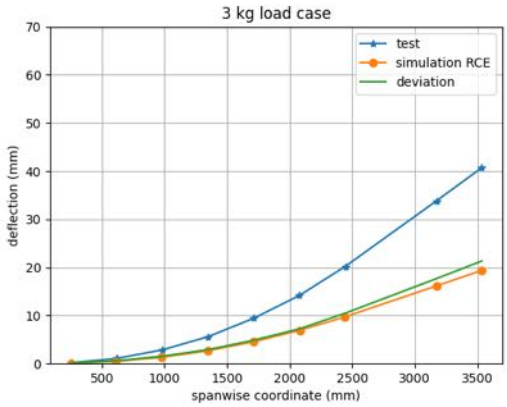


Figure 72: Span-wise bending of wing under tip load (-1 wing RCE model)

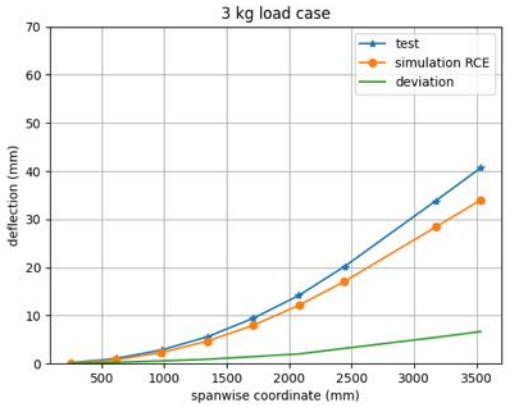


Figure 73: Span-wise bending of wing under tip load (-1 wing RCE updated model)

Nr	Mode	GVT	RCE FEM	deviation %	updated RCE FEM	deviation updated %
0	2n wing bending-s	2.94	3.81	29.68	2.89	-1.54
1	3n wing bending-a	7.57	10.05	32.82	7.8	2.99
2	wing torsion-s	10.27	12.25	19.3	10.65	3.74
3	wing torsion-a	10.73	12.7	18.33	10.86	1.24
4	4n wing bending-s	12.13	17.38	43.31	13.25	9.25
5	2n wing inplane-s	15.07	17.86	18.54	14.43	-4.27

Figure 74: Comparison of eigen frequencies of the flexible modes: GVT vs RCE FE model vs updated RCE FE model of the -1 aircraft (*i*n - *i* nodes in the mode, s - symmetric, a - antisymmetric)

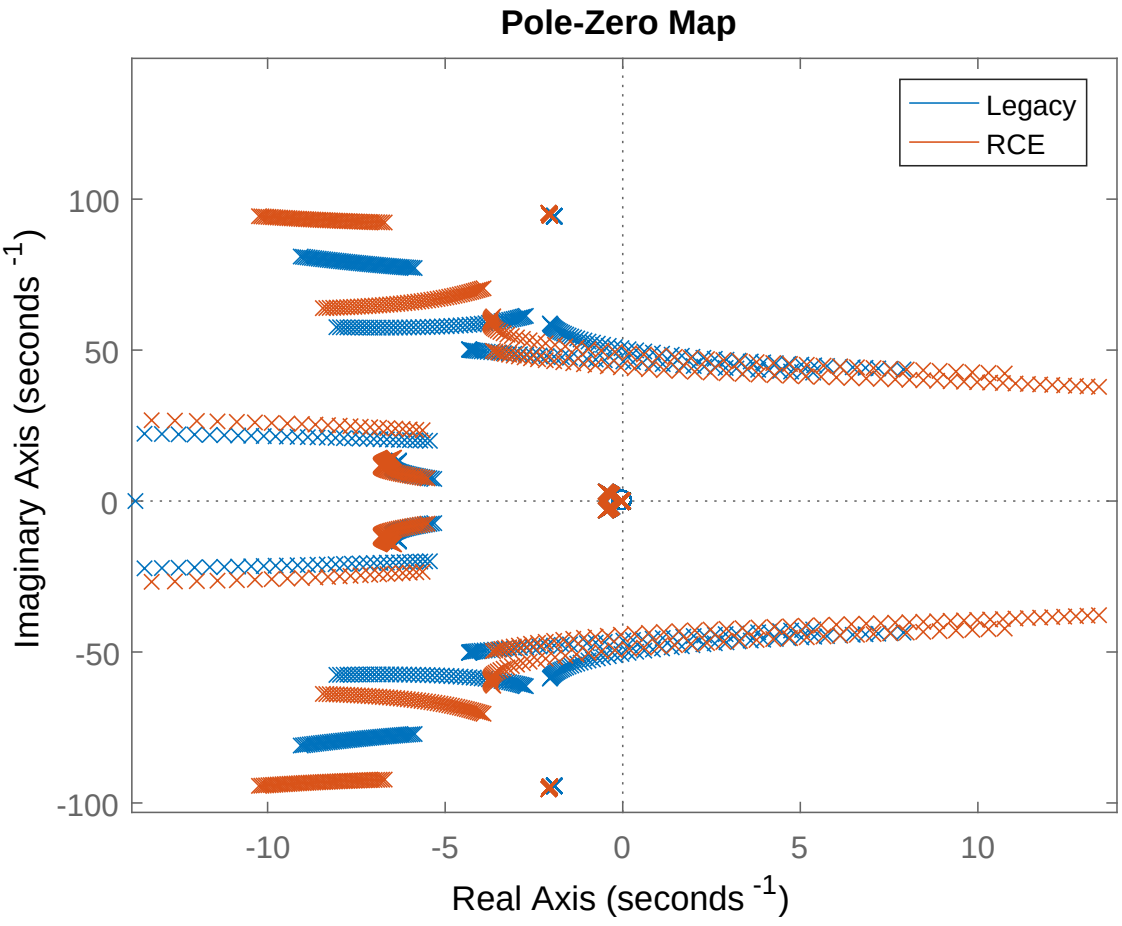
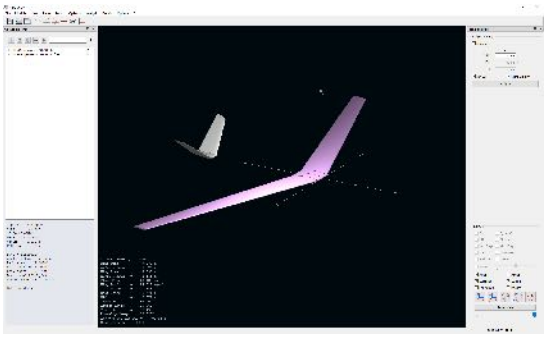
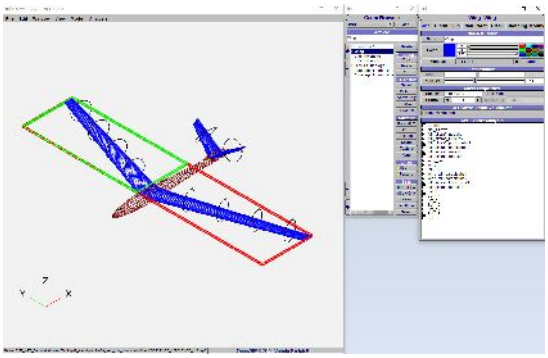


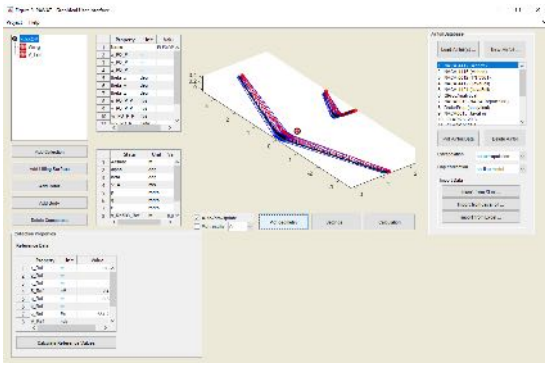
Figure 75: Comparison of pole trajectories of the ASE models: Legacy Flexop model vs RCE generated model of the -1 aircraft



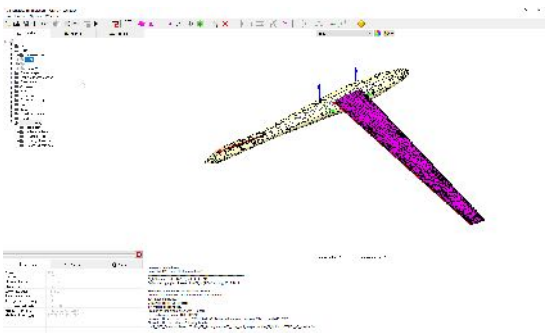
(a) XFLR5.



(b) OpenVSP.



(c) PAWAT.



(d) FlightStream.

Figure 76: T-FLEX demonstrator modelled in different tools.

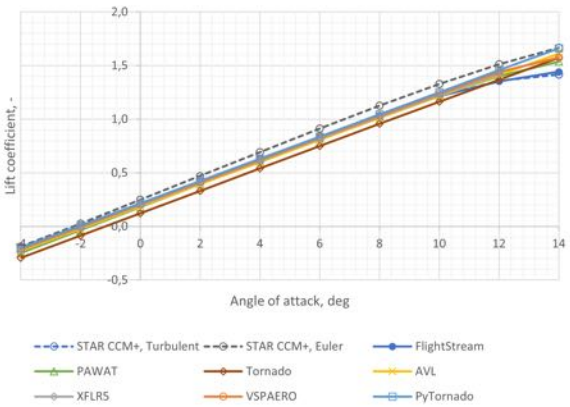


Figure 77: Lift coefficient C_L with respect to the angle of attack α.

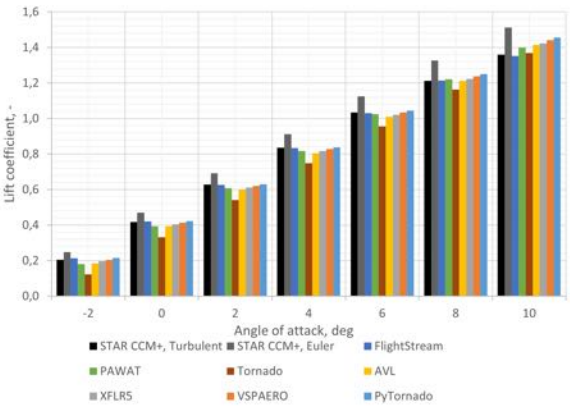


Figure 78: Lift coefficient C_L with respect to the angle of attack α.

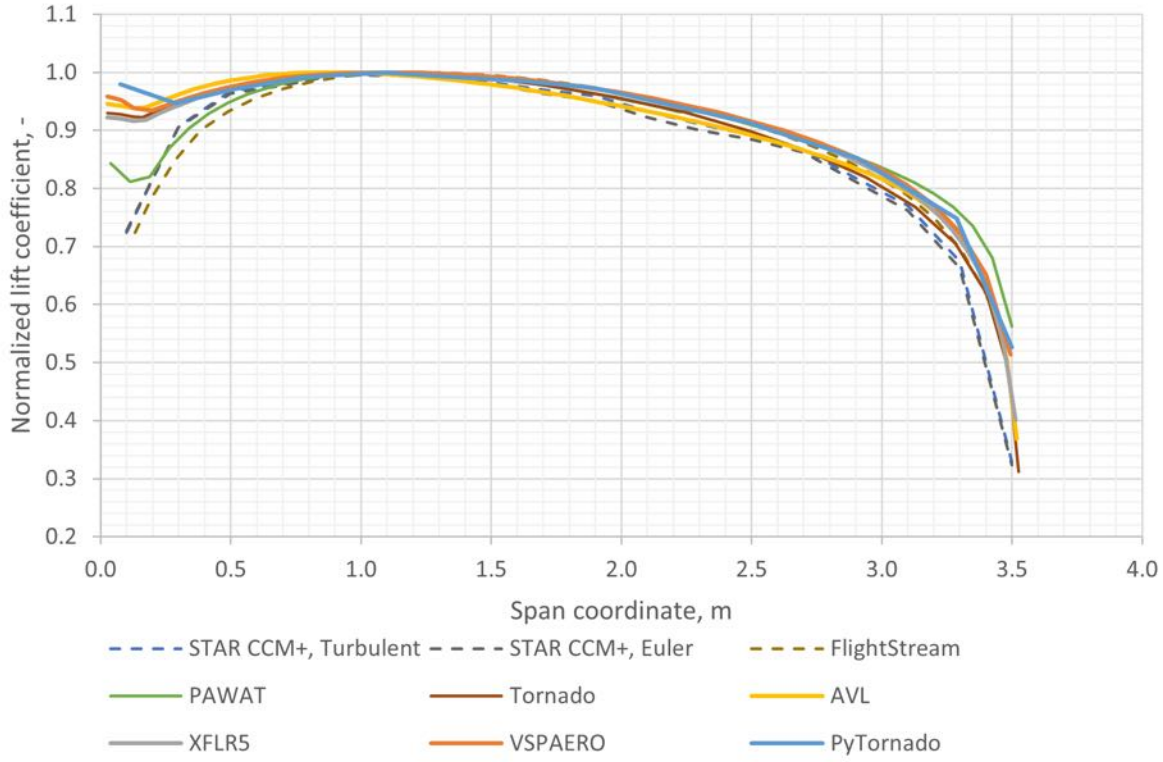


Figure 79: Spanwise normalized lift distribution for $\alpha = 2\,\text{deg}$. The local lift coefficients are normalized with respect to the maximum local lift coefficient of the same tool.

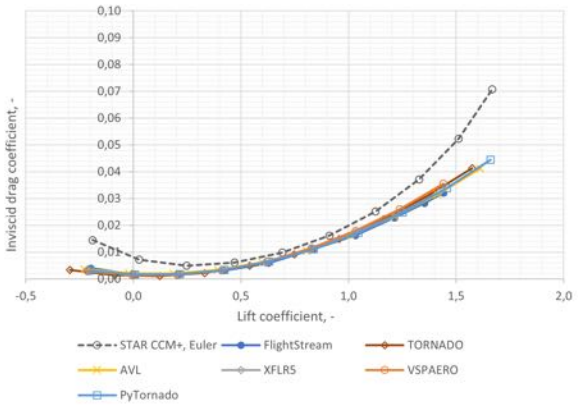


Figure 80: Inviscid drag coefficient C_{D_i} with respect to the angle of attack α.

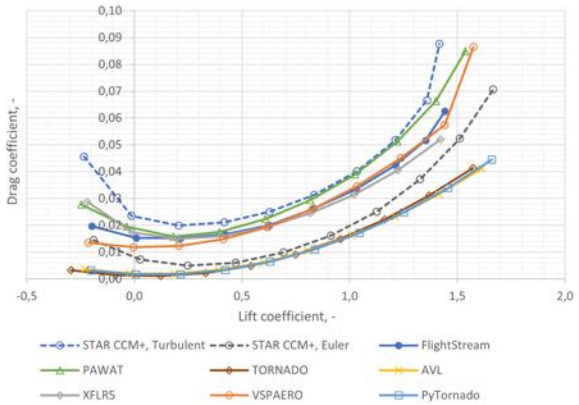


Figure 81: Total drag coefficient C_D with respect to the angle of attack α.

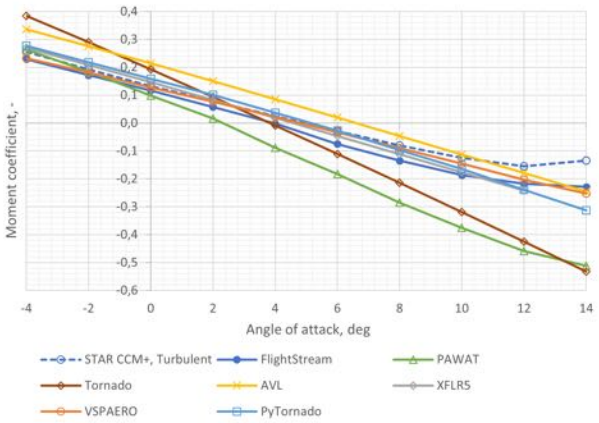


Figure 82: Pitching coefficient C_m with respect to the angle of attack α .

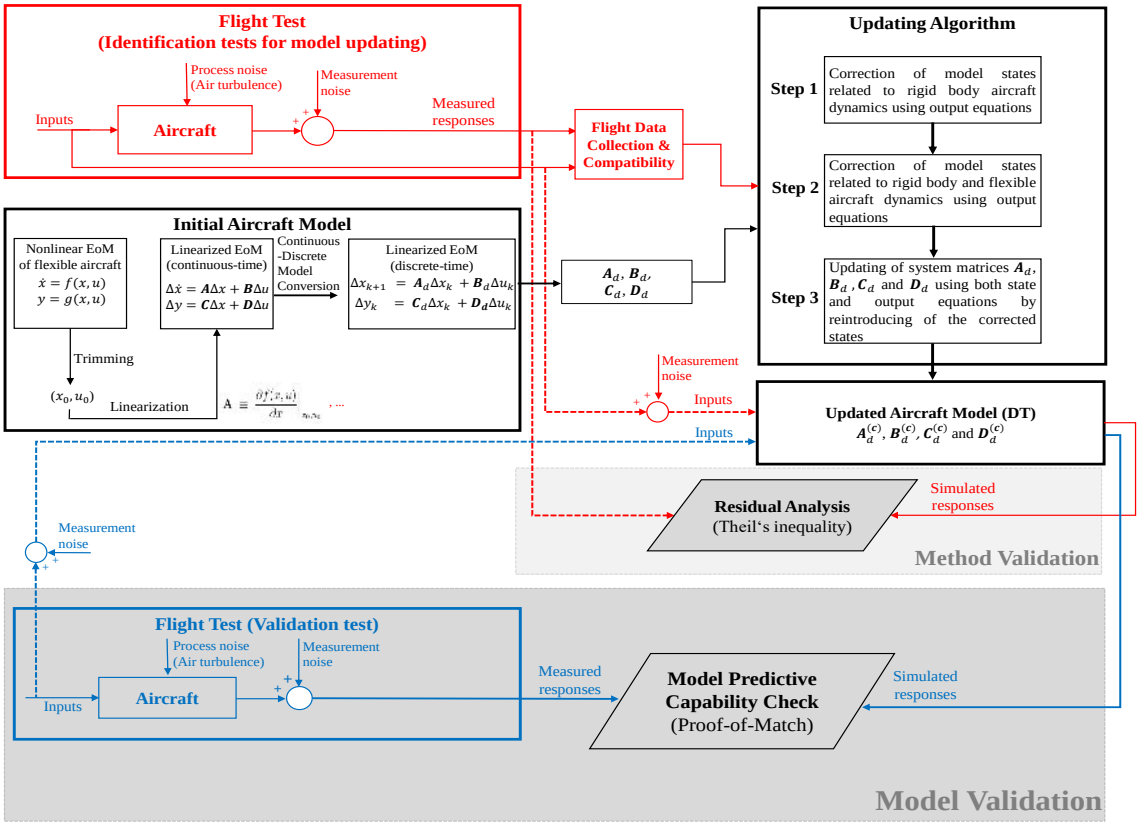


Figure 83: Overview of the model structure, updating algorithm and validation process [104]

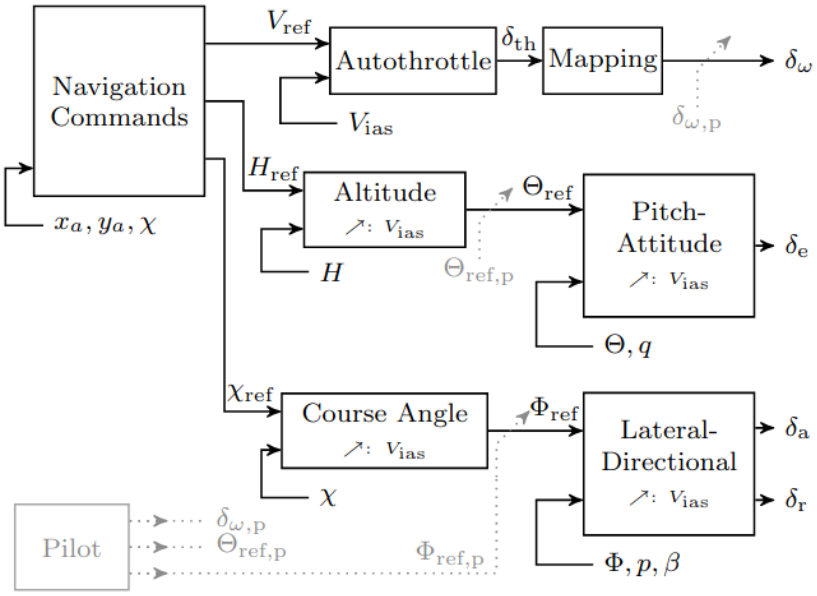
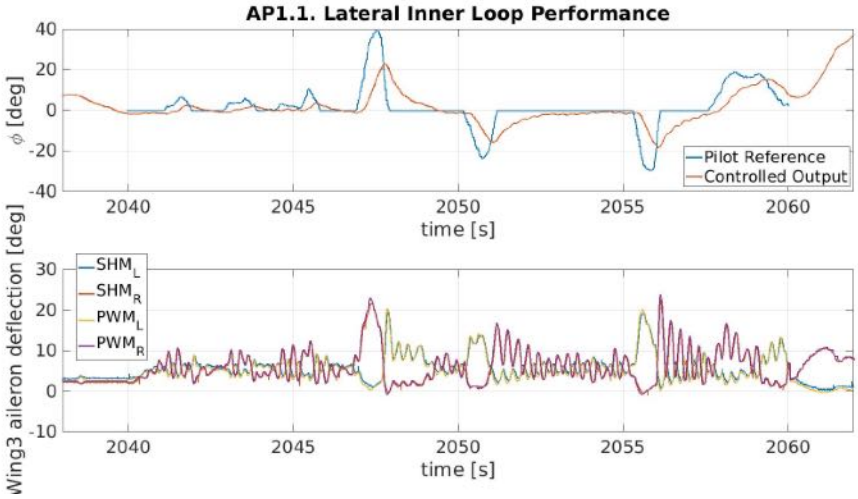
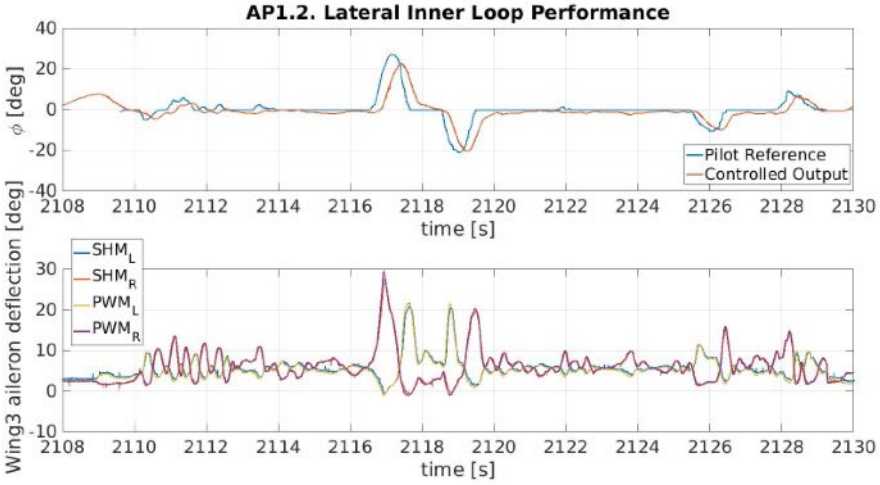


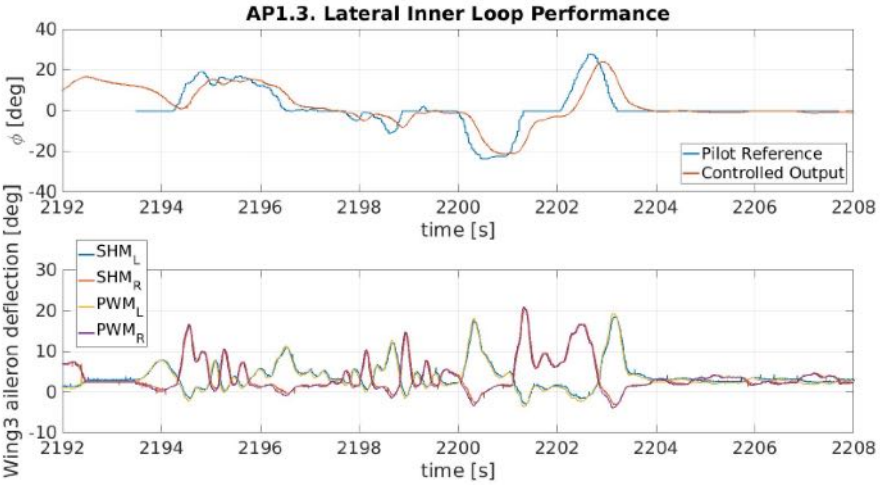
Figure 97: Structure of the baseline controller



(a) Lateral inner loop for AP1.1. - Flight Test 11



(b) Lateral inner loop for AP1.2. - Flight Test 11



(c) Lateral inner loop for AP1.3. - Flight Test 11

Figure 98: Comparison of different lateral inner loop controllers during Flight Test 11

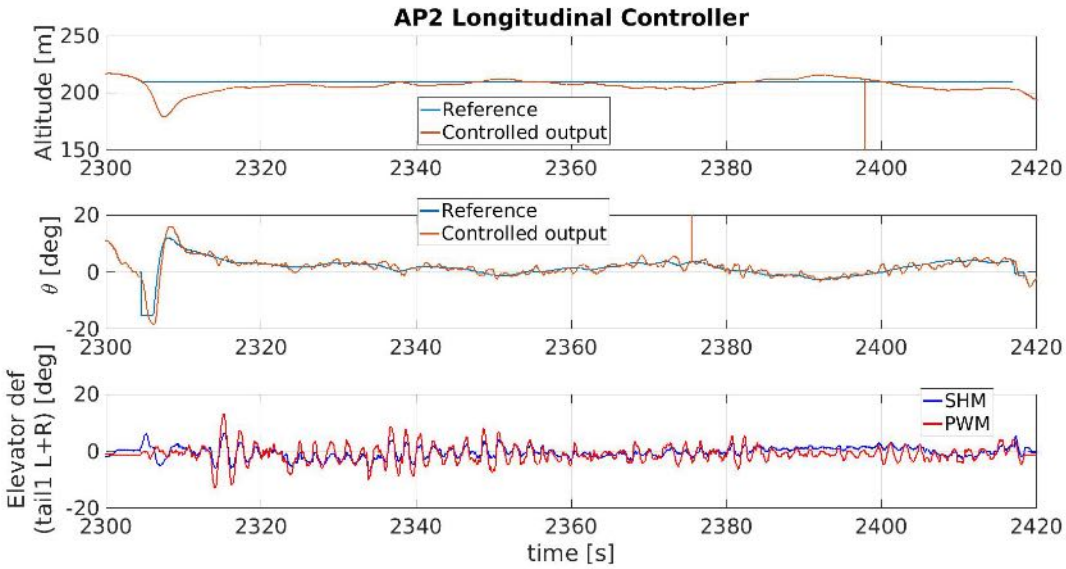


Figure 99: Flight test evaluation of the longitudinal control laws

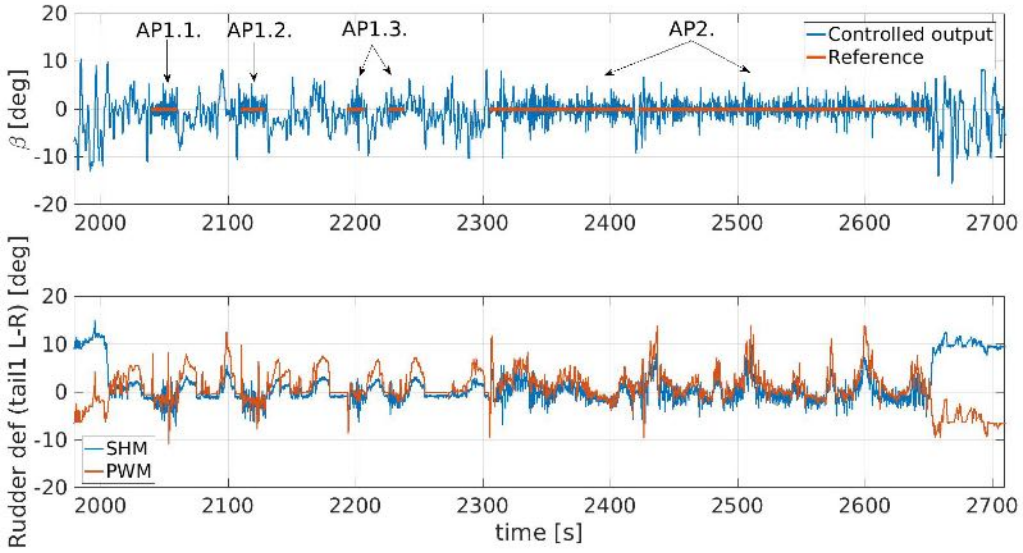


Figure 100: Sideslip loop performance during Flight Test 11

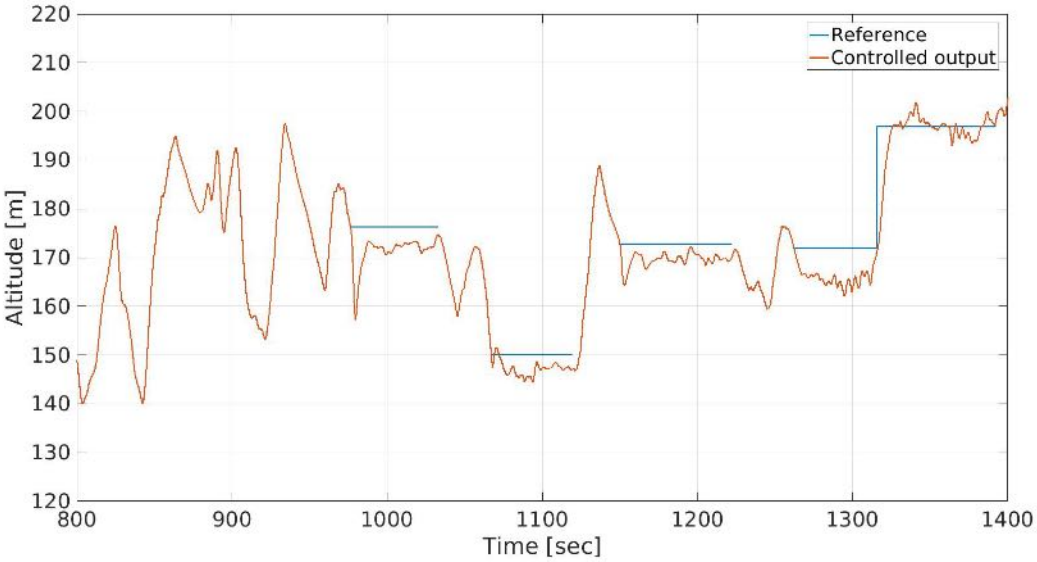


Figure 101: Altitude reference tracking during FT12

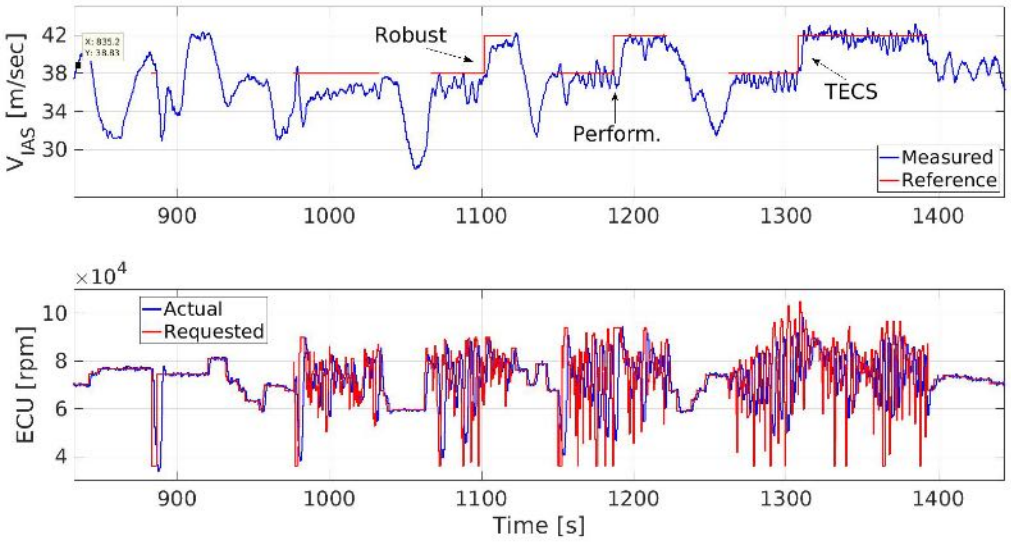


Figure 102: Comparison of different autothrottle controllers during FT12

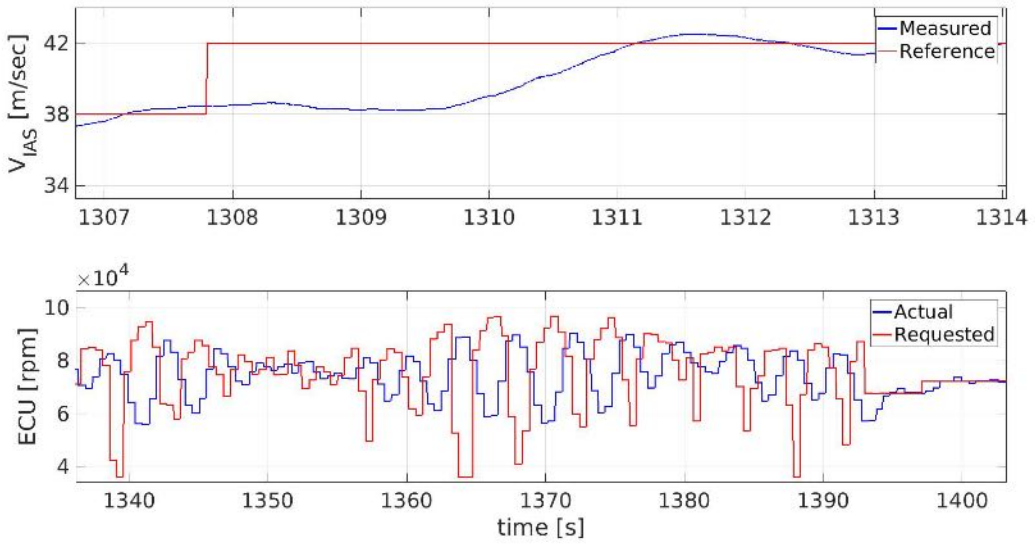


Figure 103: Speed tracking and the corresponding RPM signal

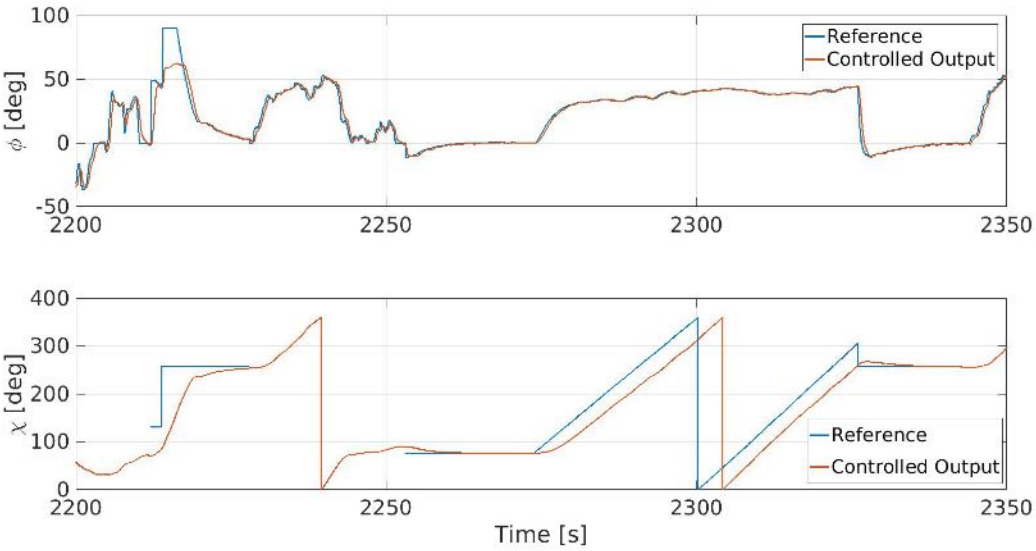


Figure 104: Course angle tracking performance during reference step change and coordinated turn

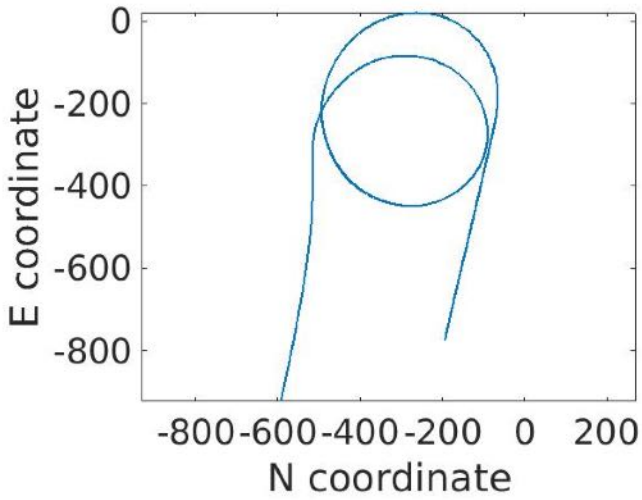


Figure 105: Coordinated turn

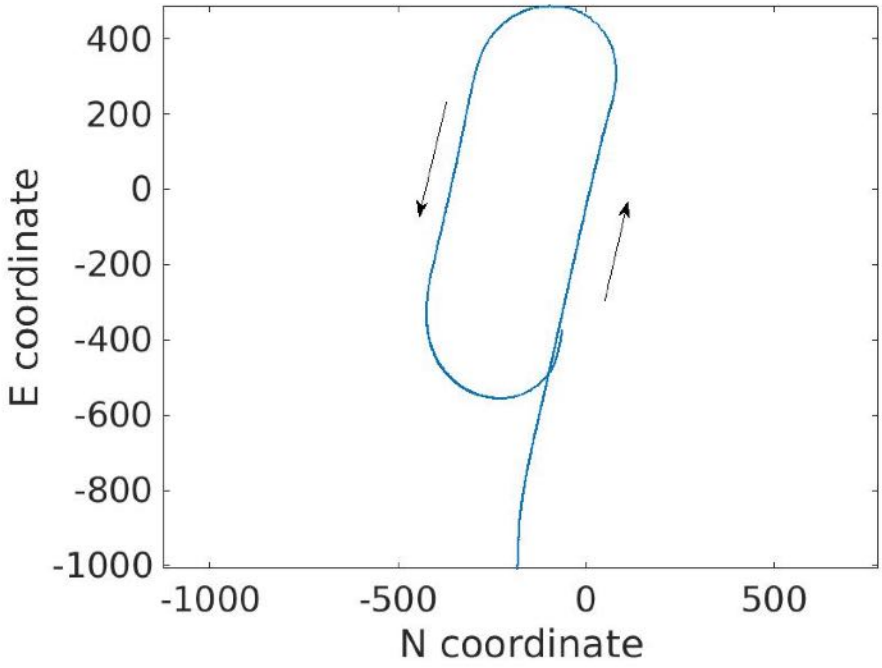


Figure 106: Horserace flight pattern

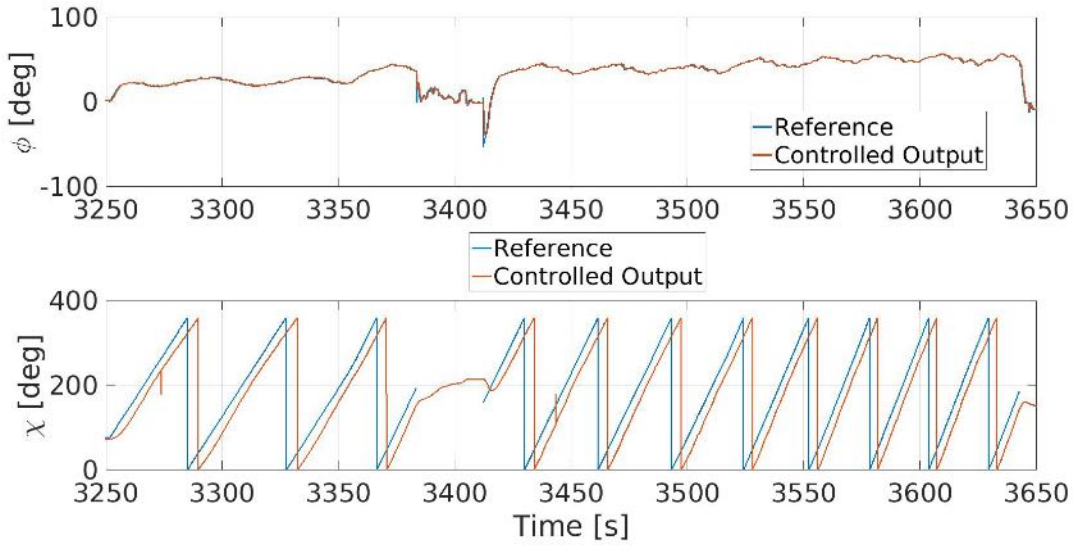


Figure 107: Course angle for full circle tests with increasing speed during FT16

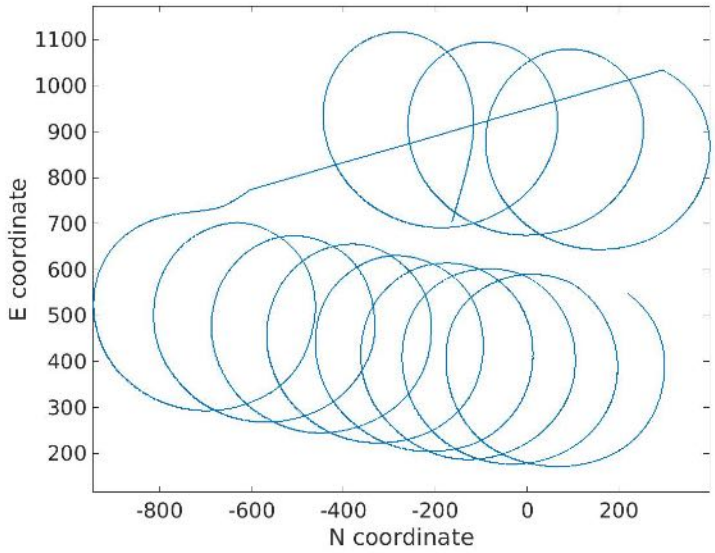


Figure 108: Full circle trajectories with increasing speed during FT16

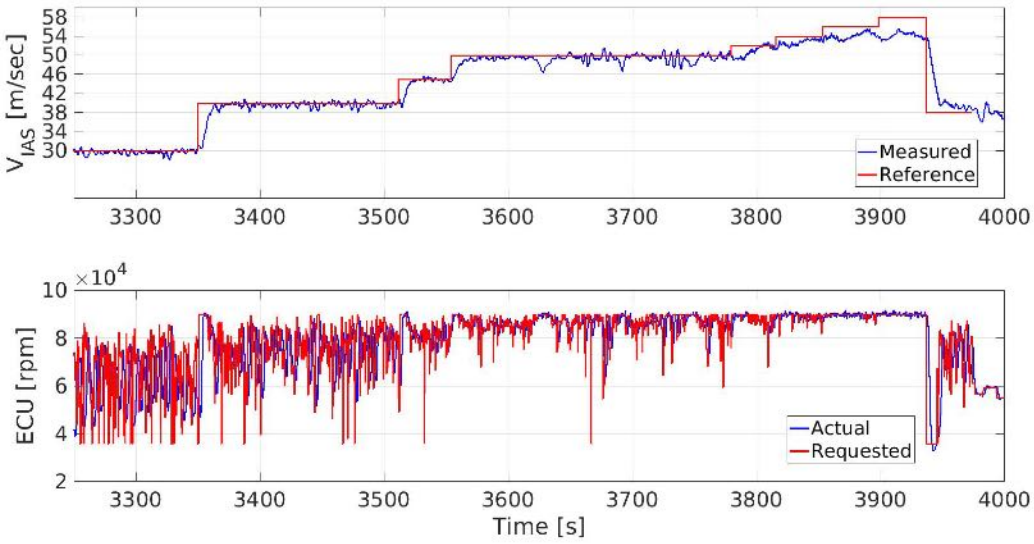


Figure 109: Increasing speed during FT16

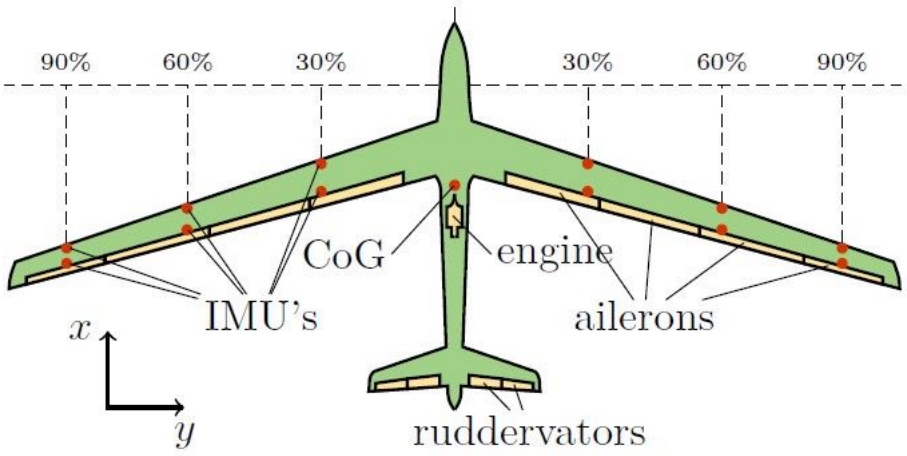


Figure 110: Demonstrator control surfaces and IMU locations

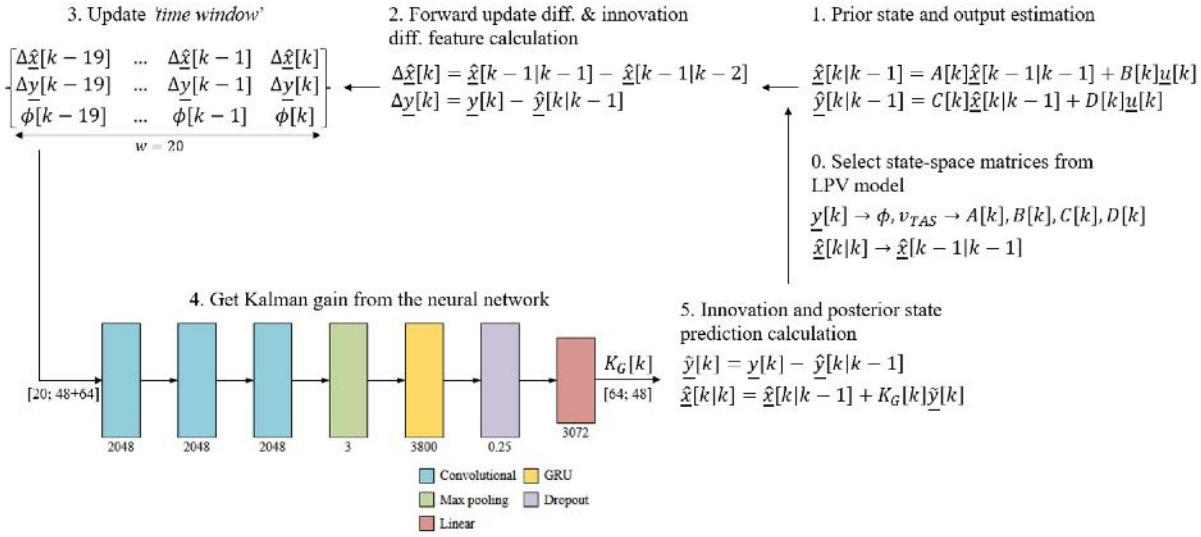


Figure 111: KalmanNet pipeline

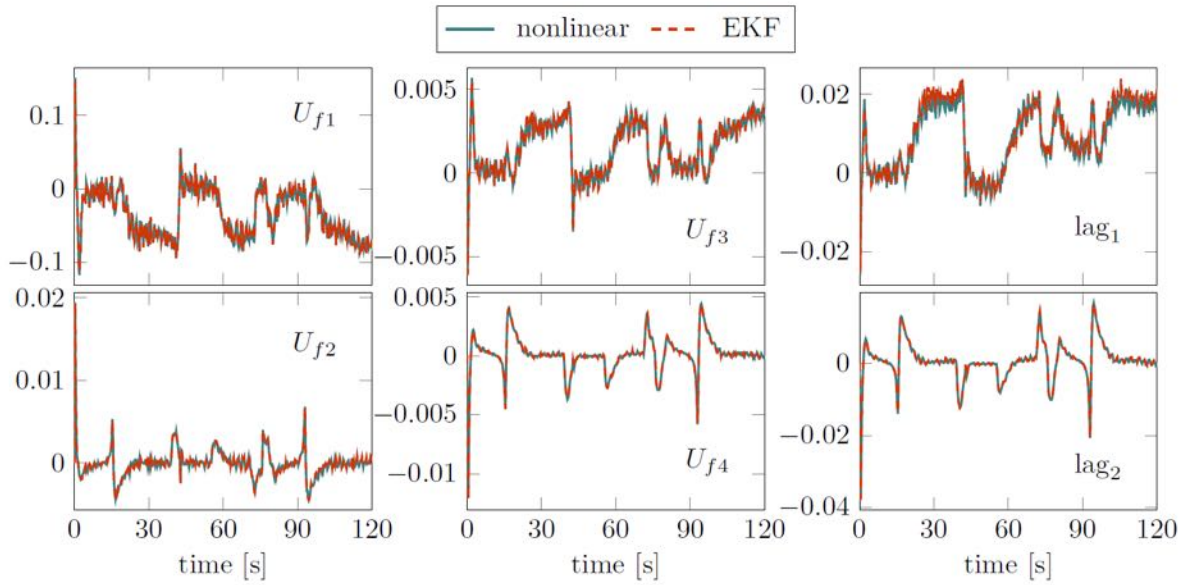


Figure 112: LPV-based EKF results

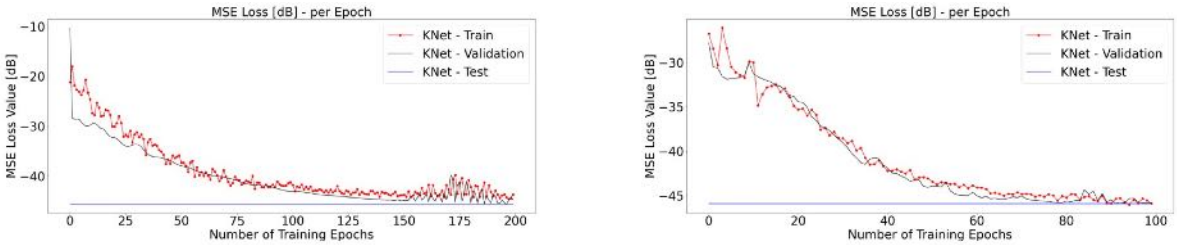


Figure 113: Linear (*left*) and convolutional (*right*) architecture training graphs

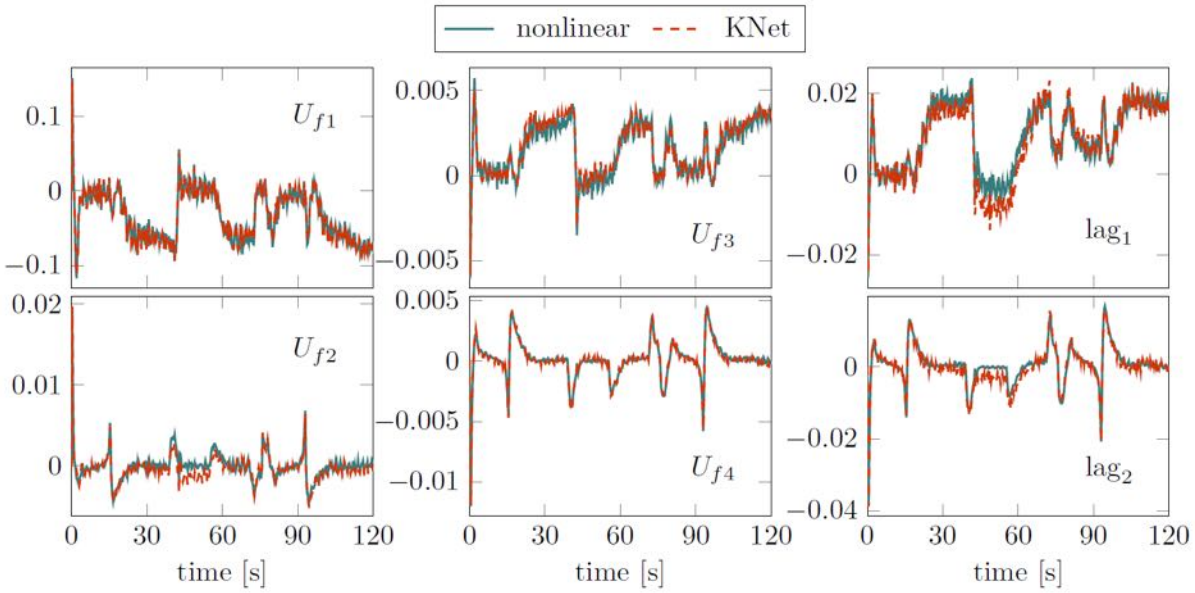


Figure 114: KalmanNet results with linear architecture

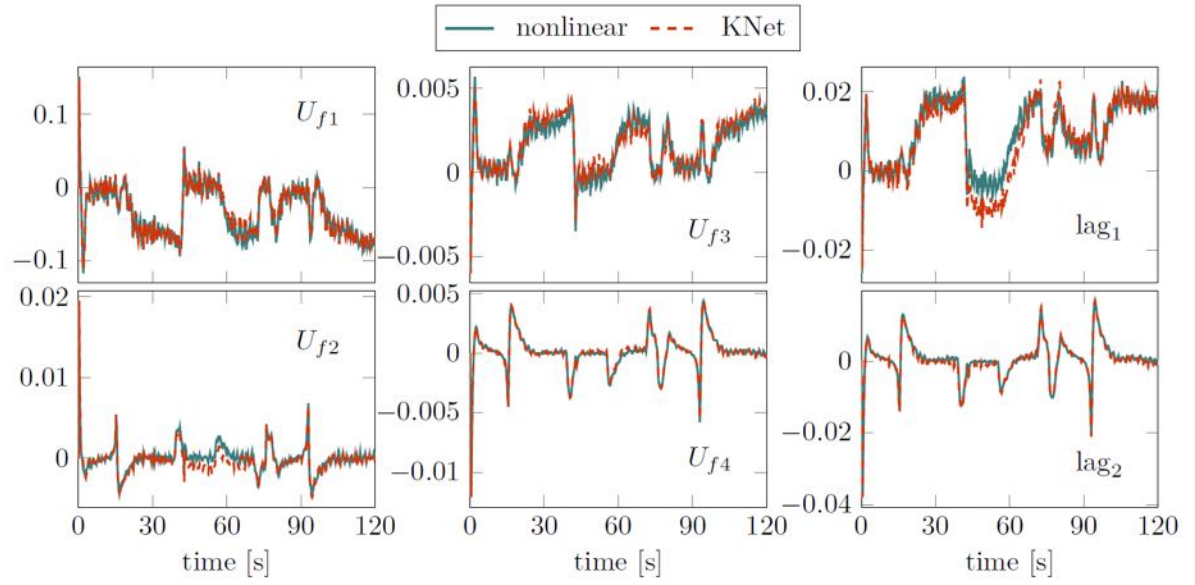


Figure 115: KalmanNet results with convolutional architecture

Node	Content
?-? xml	version="1.0" encoding="utf-8"
cpacs	all(header, vehicles?, missionDefinitions?, airports?, flights?, airlines?, studies?, toolspecific?)
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 116: CPACS dataset of demonstrator workflow

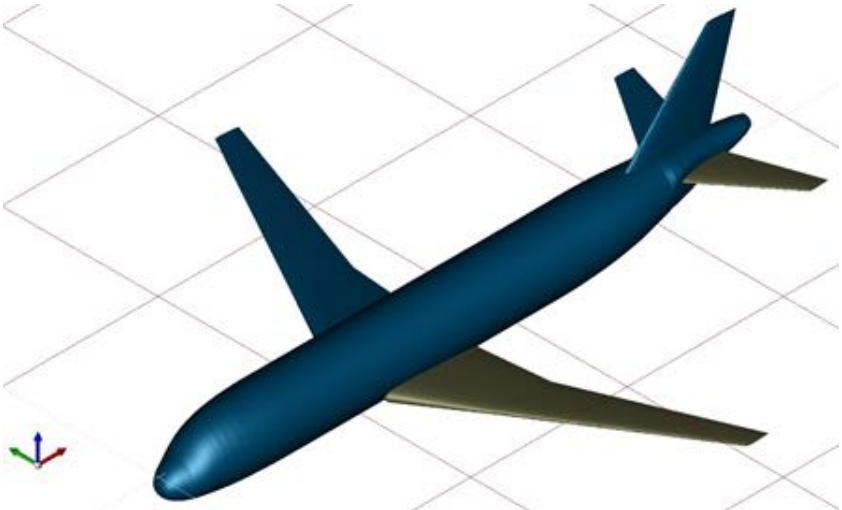


Figure 117: Outer geometry of the FLIPASED DLR-D150 configuration generated from the CPACS dataset using TiGL Viewer 2.1.3

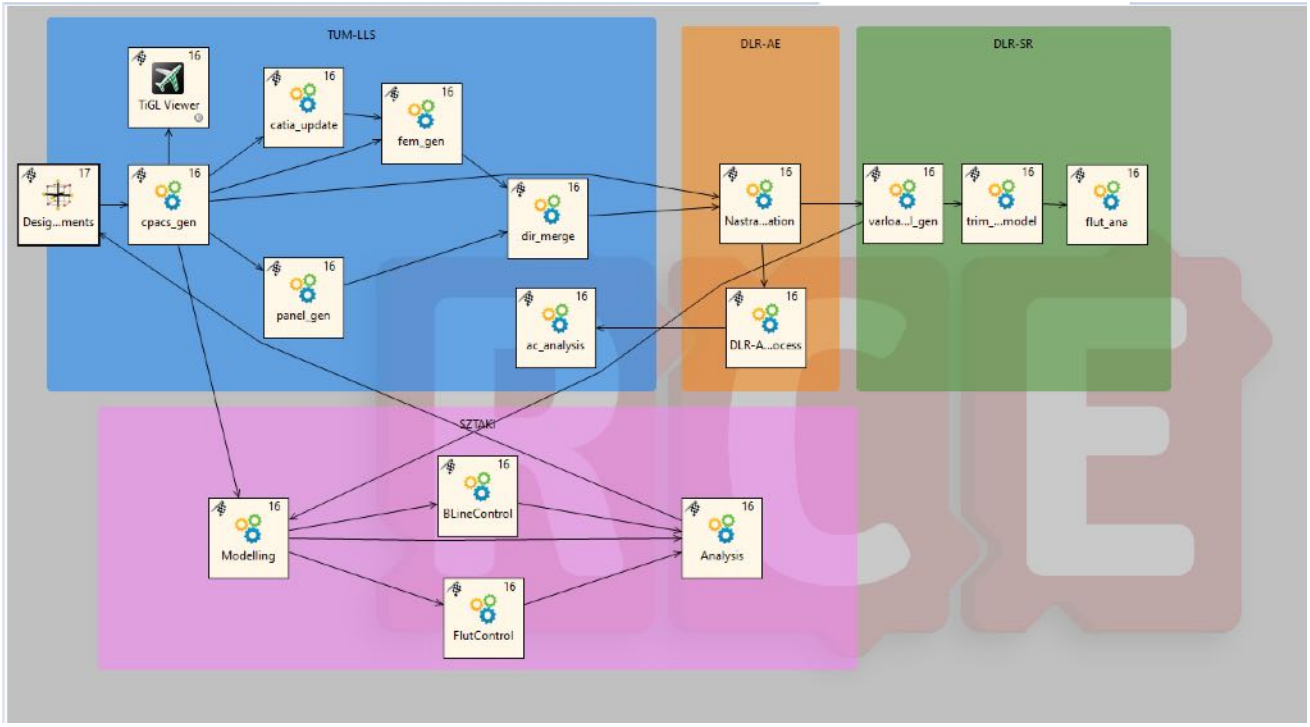


Figure 118: RCE tools in the NASTRAN aeroelastic model integration block

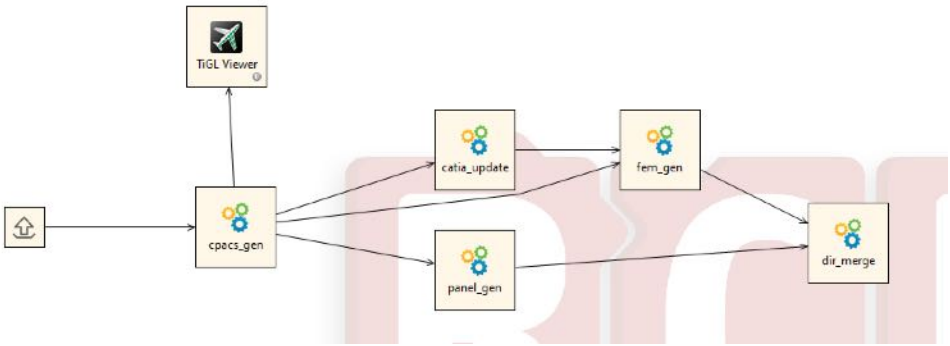


Figure 119: RCE tools in the Aircraft model generation sec

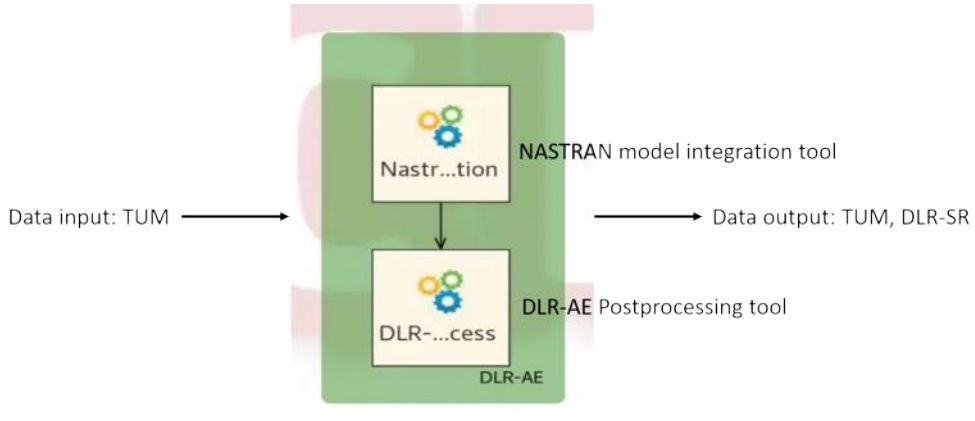


Figure 120: RCE tools in the NASTRAN aeroelastic model integration block

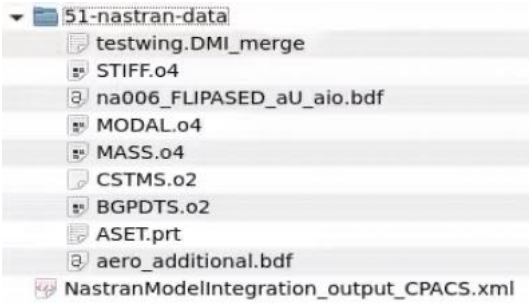


Figure 121: Output files and format from NASTRAN aeroelastic model integration tool

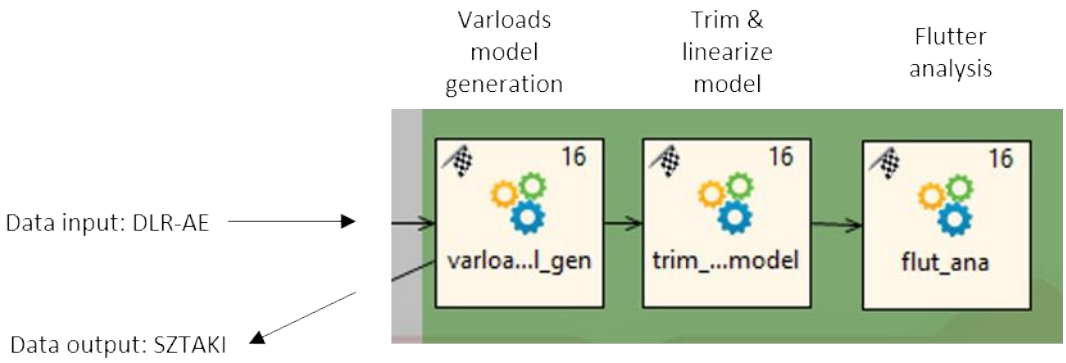


Figure 122: RCE tools in the ASE model integration block

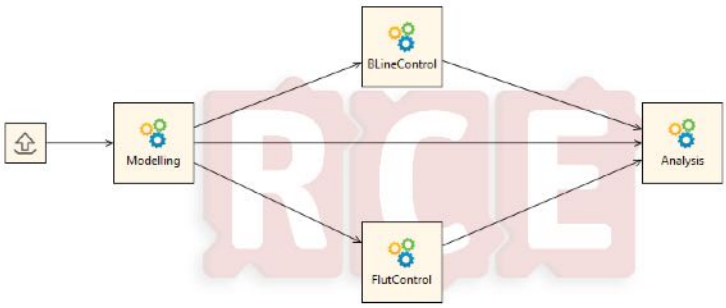


Figure 123: RCE workflow for the aeroelastic model reduction for flutter suppression control design

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.

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 125 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.

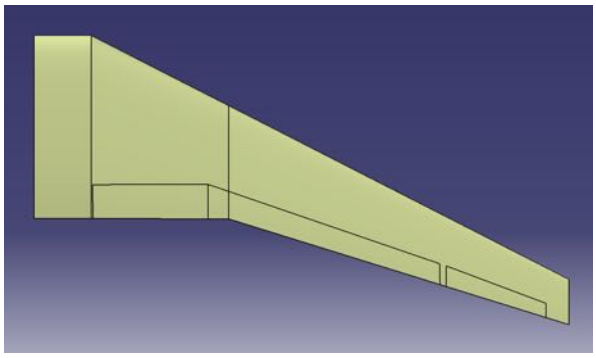


Figure 124: D150 initial control surfaces

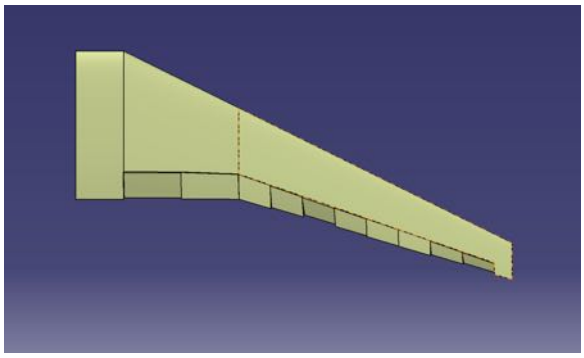


Figure 125: D150 modified control surfaces

As depicted in the figure 124, 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.

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 [56].

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 aerodnymic model in NASTRAN
- Definition of loadcases considered and the down-selected loadcases used in the structural optimization

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.

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 $P_{c,\mathrm{mx}}$ is of special interest. The loads are estimated with the force summation method (FSM)

$$P_c = T_{cg}\left(P_g^{\mathrm{ext}} - P_g^{\mathrm{iner}},\right) \tag{26}$$

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 [9, 50].

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 $\dot{U}_{z,t}(t)$

$$\begin{aligned} U_{z,t}(t) &= \begin{cases} \frac{\bar{U}_t}{2}\left(1-\cos\left(\frac{\pi}{H_t}\left(U_\infty t - x_z\right)\right)\right), & \text{if } \frac{x_z}{U_\infty} \le t \le \frac{2H_t + x_z}{U_\infty} \\ 0, & \text{otherwise} \end{cases} \\ \dot{U}_{z,t}(t) &= \begin{cases} \frac{\bar{U}_t \pi}{2H_t} U_\infty \sin\left(\frac{\pi}{H_t}\left(U_\infty t - x_z\right)\right), & \text{if } \frac{x_z}{U_\infty} \le t \le \frac{2H_t + x_z}{U_\infty} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{27}$$

The maximum gust intensity and gust half length are $\bar{U}_t$ and H_t [24]. With evolving time t the aircraft flies through the gust from nose to aft. This is shown in Figure 144. The aerodynamic model of the aircraft 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

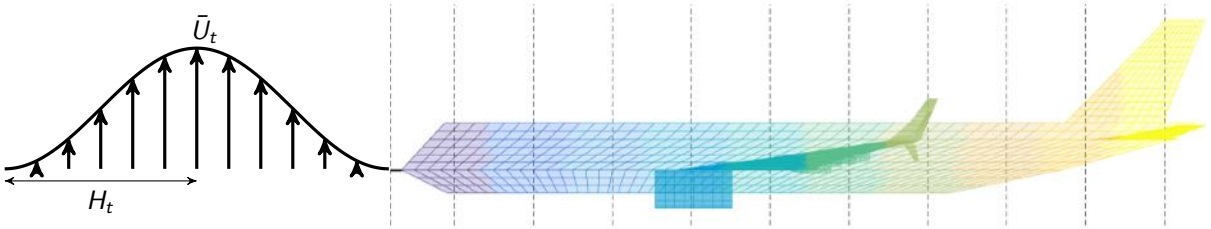


Figure 126: 1-cosine gust and aircraft gust zones.

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 [100]. 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 [44]. 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

$$G_{z,d}(s) = e^{-t_{z,d}s}, \tag{28}$$

where $t_{z,d}$ is the time delay in seconds and s is the Laplace variable [44]. A second-order Padé approximation of a time delay is

$$G_{z,d}(s) \approx \frac{s^2 - \frac{6}{t_{z,d}}s + \frac{12}{t_{z,d}^2}}{s^2 + \frac{6}{t_{z,d}}s + \frac{12}{t_{z,d}^2}}. \tag{29}$$

It converts to a linear state-space system [34]. 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.

GLA controller design The GLA control is synthesized based on model predictive control (MPC). Figure 127 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 $k + n_p$ [8]. MPC then applies the first predicted control input increment. A time step of Δt_s later, the optimisation is repeated [94].

For GLA the elevators and ailerons on both wings are used. In Figure 452 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) $P_{c,\mathrm{mx}}$ is estimated based on the given measurements.

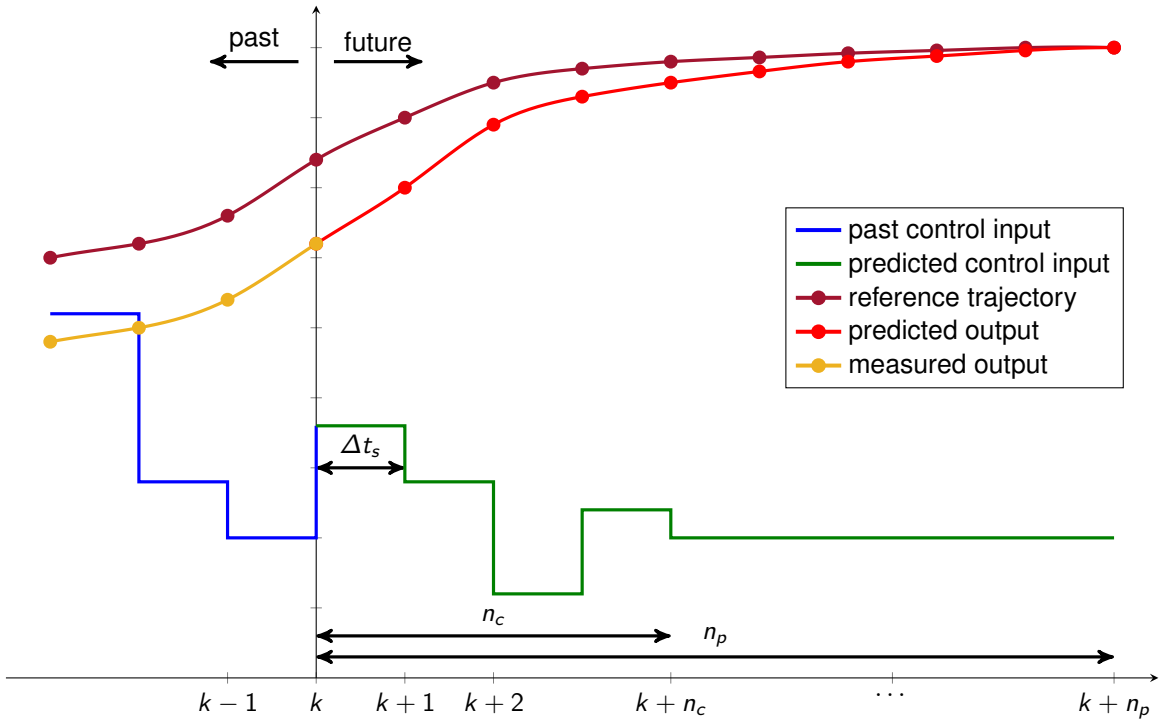


Figure 127: MPC principle [94].

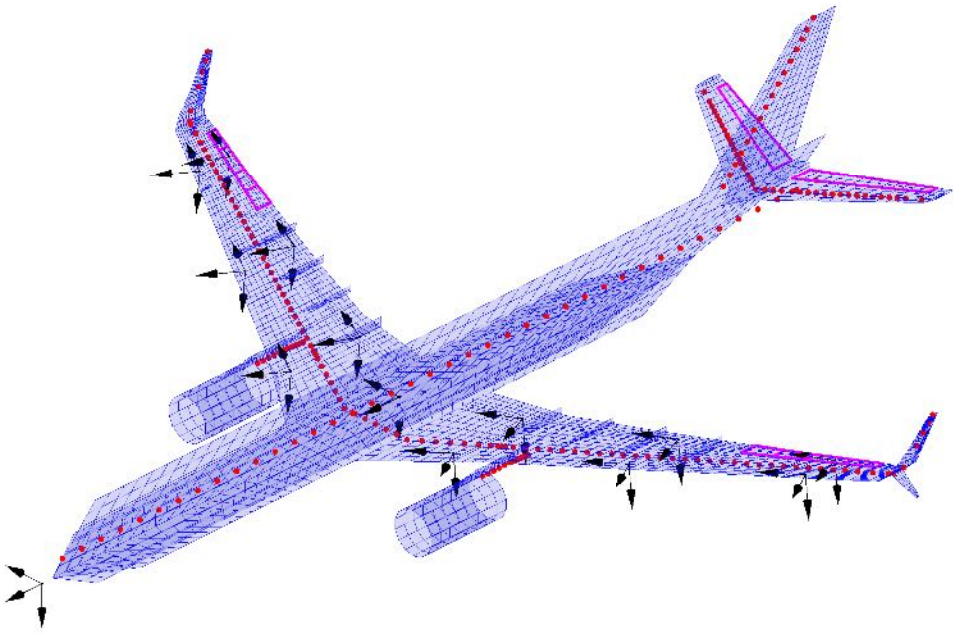


Figure 128: Reference flexible aircraft model defined by the structural grid (red), the aerodynamic panel model (blue), the deployed control surfaces for GLA (magenta) and the sensor coordinate system locations and orientations (black).

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 weather 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 the open loop model. The LTI model of the aircraft is 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.

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

Trefftz plane implementation in NASTRAN The Treffz 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.

VLM-based near-field implementation A vortex lattice method (VLM) - based near-field implementation [49] was studied as a candidate tool in this exercise. The work presented in [49] extends the classical VLM implementation in the loads environment VarLoads [37] 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 2.1.3.

PANUKL-based drag estimation tool PANUKL is a software package to compute the aerodynamic characteristics of an aircraft using low order panel methods [30]. 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 2.1.3.

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.

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

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 17. 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 17.

Table 17: 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.

2.1.4 Deviations, their reason, impact on the project and corrective actions

The project is heavily impacted by the COVID related restrictions, what are even more striking in WP1, since both the hands-on work on the demonstrator must be postponed several times and the supporting teams of DLR and SZTAKI were only able to be on site at TUM for a very limited time.

On the other hand, this facilitated the need of online collaborative tools and methods. What has been established on several fronts: the teams are using common software development repositories using the SZTAKI hosted Gitlab site. The teams also collaborated more closely on developing tools compatible with the CPACS/RCE framework, what can be integrated into the workflow remotely.

Task 1.1: Requirements Capture is mostly done, but on-site brainstorming sessions would highly facilitate the discussions. The team adopted a weekly webex session where dedicated sessions are devoted to requirement capture.

Task 1.2 A/C Reference Model Definition – the team selected a suitable aircraft benchmark, the D150, which is well known and understood by DLR and its limitations are set, to limit the scope of the consortium. The deliverable related to this task (D1.5) was delivered late, but the actual work and decision within the consortium was done on time, and this does not have impact on the critical path of the project.

Task 1.3 Collaborative Work Process Definition – based on the CPACS and RCE standards the work process is defined but there is significant delay in the integration of these blocks, since many partners are permanently at home office, where they cannot access the company's main computer infrastructure.

2.2 Explanation of the work carried out per WP - Work Package 2

2.2.1 Objectives and activities

Within Work Package 2, the driving objective is to address the feedback control functions construction. The main objective of the WP is to develop a bundle of functions allowing to design the control functions in an automated manner, in order to be included in the global Multi Disciplinary Optimisation (MDO) process. Since all control design algorithms are model based, developing control oriented models for each control design is one of the key tasks of the WP. This MDO being the central objective of FLiPASED, the proposed process should fit this frame and should not involve that much external user intervention. This is why a strong attention in constructing systematic approach is given. Finally, the developed methods and tools need to be validated by ground and flight test data. This WP involves three research groups, the DLR, ONERA and SZTAKI.

As a sub-objective, one seeks for the development and maturation of tools used for structural design, aerodynamic design and aeroelastic design. The second sub-objective objective set concerns the development and integration of tools such as control, detection and estimation synthesis into the MDO toolchain. The third sub-task is to utilize wing shape control for minimization of induced drag, based on which the new advanced FliPASED wing is designed. The fourth sub-task is the development of analytical redundancy methods for fault detection and isolation (FDI) for flexible aircraft. The fifth sub-task is the validation of data science based methods for modelling and control of flexible aircraft.

2.2.2 Starting point and approach

Overall control big picture

The MDO loop of the WP considers the constuction of dynamical models of the aircraft, sensors and actuators. These models, together with objectives and performance specifications are the starting point of the control design. Generally, aircraft manufacturers control design workflow follows what we can call a frequency grid approach. This approach consists in designing different controllers, through a frequency guideline. Each of them then address a single phenomena an aircraft is faced during its operation. Within the overall MDO process philosophy, and in this WP, we aim at following this very same approach. With reference to Figure 129, one may notice that different phenomena (flight, loads...) usually occurs at different frequencies. These frequencies are dependent on the geometry and structure of the aircraft, and in the considered case, one may expect even more blending in the phenomena. Still the big picture remains. This sequential control structure will be kept in mind in the WP2 flow to stick to industrial and practical expectations.

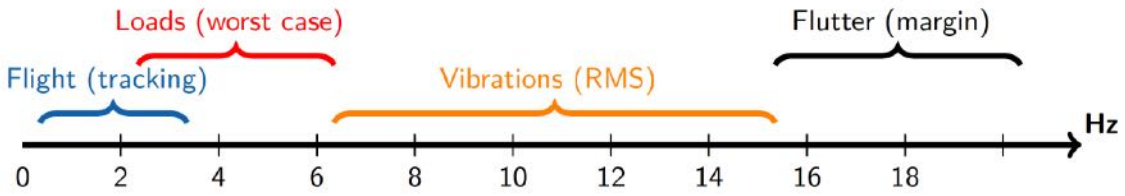


Figure 129: Frequency grid of the physical phenomena occurring over an aircraft. Ranges and values are different from an aircraft / geometry to an other

All four, flight, maneuver, gust load and flutter controllers were considered and a preliminary control architecture has been deployed. Without entering into much details, this is then presented in the following section. Moreover, as an indistinguishable point, model approximation and analysis are also considered in these tasks. In addition to the four controllers discussed above, a static scheduled aileron deflection for drag minimization is also investigated in the project.

Connection between the MDO unicorn with FLEXOP

From the **modelling** point of view, the initial starting point was the geometrical, structural and aerodynamic model of wing -1 which is the legacy of FLEXOP project.

In order to achieve a fully automated MDO toolchain, a parameterized geometrical model is the cornerstone of the whole toolchain. All the down-streaming FE-Model and aerodynamic model would be build upon it. Based on the available wing -1 Catia model, it would be parameterized with wing planform parameters (sweep angle, span and taper ratio), structural layout parameters (spar position, jig twist) and control surfaces design parameters (flap position). To have a better drag reduction effects with control surfaces deflection, the number of flaps is increased. It will give more freedom for control law designer. To add drag estimation functionality to the toolchain, a suitable aerodynamic solver needs to be chosen and coupled with Nastran.

There are lot of different tools involved in the model generation process. To avoid the human intervention in the MDO toolchain, interface between software needs to be defined specific and all the human operation needs to be programmed or recorded in Macro. Besides, component model of aircraft would be delivered separately by partners and would be assembled eventually. To automatize this process, detailed interface definition was carried out among partners.

From the **control** design point of view, the starting point is from previous research projects, especially FLEXOP. These methods and algorithms need to be adopted to the MDO toolchain, which requires a

special attention. For example, the control oriented modeling involves some heuristic steps. These steps need to remain robust for model variation due to the MDO optimization. A possible way for such adaptation is the automatic evaluation of the accuracy of the resulting low order model and automatic increase in the states for the low order model to maintain sufficient accuracy. In case of the control design blocks, the control performance specifications need to be adopted to the MDO toolchain in a way that they can be automatically relaxed in case no feasible controller can be found.

2.2.3 Efforts and achieved results, name of involved partners

The description of the efforts and achieved results are presented in the following groups:

- MDO toolchain description;
- Control oriented model development for the various control algorithms;
- Automated control design algorithms;
- Tool implementation and validation;
- Wing shape control based drag optimization and design of the new advanced FLiPASED wing;
- Analytical redundancy methods for FDI of flexible aircraft;
- Validation of data science based methods for modelling and control of flexible aircraft.

Results and validation of the integrated design toolchain (DLR, TUM, SZTAKI, ONERA)

The objective of this section is to conduct a comprehensive comparison of results and findings obtained from various sources. The primary purpose of this specific assessment is to instill confidence in the developed tools and methods for the collaborative design toolchain. The data acquired from flight tests will serve as a reference point for validating models related to structural dynamics, aerodynamics and controls. To facilitate this validation process, analysis tools will be designed for test and simulation results. For the structural dynamics tools will be available in Nastran which can be tuned based on available ground testing data. For the aerodynamics the output of different tools will be compared with flight test data gathered for the demonstrator aircraft. Furthermore, the results of the different methods available for flutter analysis will be cross-compared in order to gain confidence in the aeroelastic modeling at different stages of the collaborative design toolchain.

This section provides an overview of the MDO toolchain and the tools utilized within it. Each block of the toolchain is briefly introduced, along with an explanation of the interconnection between the MDO, Hardware-in-the-Loop (HIL) test, and flight test toolchains. Finally, the results of a case study carried of with the MDO toolchain is given.

Overall architecture and tools of MDO toolchain

The central component in this scenario is the MDO toolchain, which encompasses its own optimization process and returns to the Common Parametric Aircraft Configuration Schema (CPACS) generation block after each iteration. The primary objective of the MDO toolchain is to demonstrate the enhancements achieved through optimization, encompassing aircraft geometry, sizing, modeling, and control design concurrently, in comparison to the reference aircraft. The overall architecture of the MDO toolchain is illustrated in Figure 130.

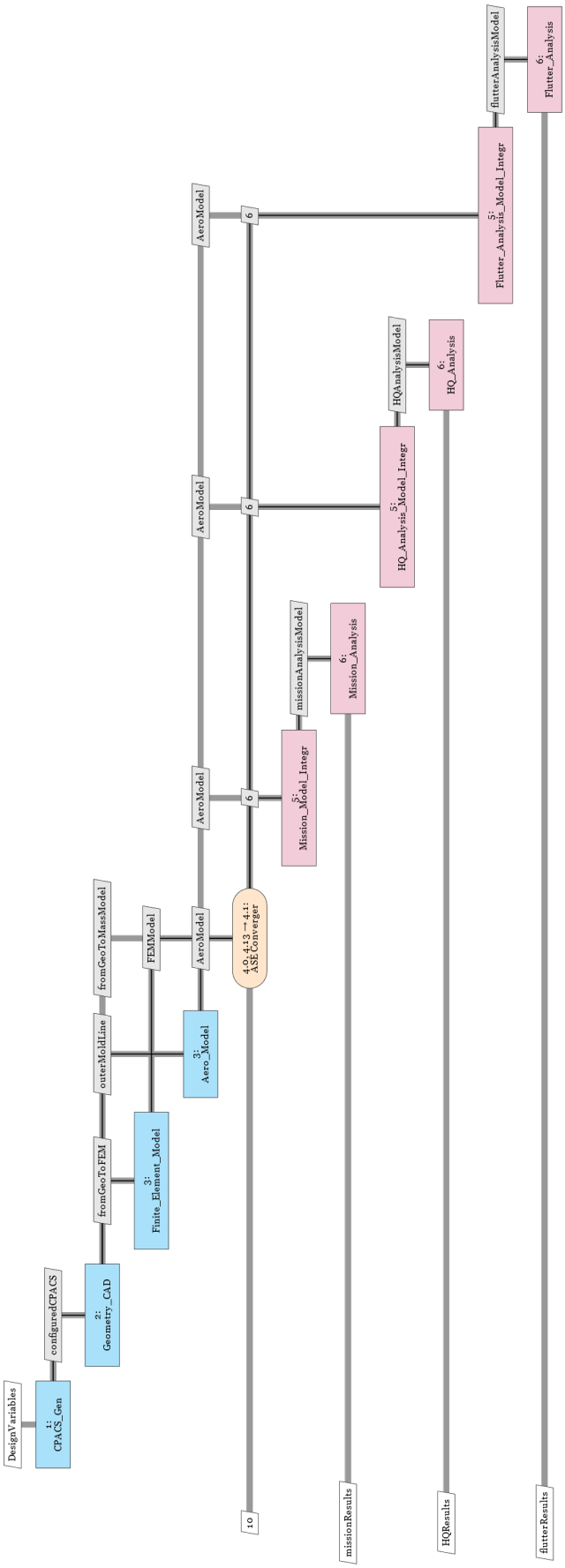


Figure 130: MDO Toolchain for demonstrator T-FLEX

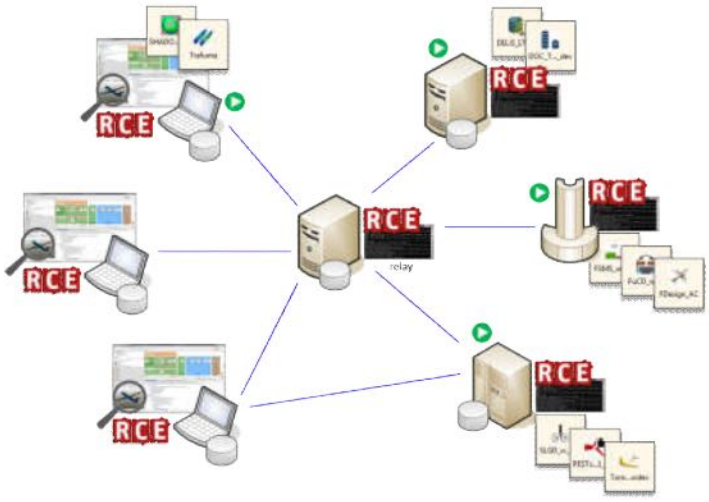


Figure 131: Distributed RCE workflow

The subsequent sections will provide a concise overview of the function blocks within the MDO toolchain and the standard tools employed. For more detailed information, please refer to the previous deliverables from the project, specifically D1.2, D1.4, D2.2, and D4.1.

Remote Component Environment

DLR's Remote Component Environment (RCE) [10, 26] is an open-source software environment for defining and executing workflows containing distributed simulation tools by integrating them into a peer-to-peer network. The RCE framework is used as the backbone for the MDO toolchain implementation. The following description has been taken from the related publication by the main developers, Boden et al. [10, 26]. RCE is being developed primarily by DLR and has been used in various engineering projects, including several aerospace projects dealing with MDO and multidisciplinary analysis (MDA). RCE has several advantages that can help to achieve more reusable multidisciplinary processes. The workflow is composed of built-in and user-defined components. Disciplinary tools are integrated as standalone components, with defined inputs and outputs, and then distributed over the network. While executing the workflow, data dependencies between the components are automatically detected, and a component is executed as soon as all its input data is available. Thus, multiple components can run at the same time. The components of a multidisciplinary process can also be executed in a distributed manner, where the tools are located on different machines with possibly different operating systems. Once configured, the peer-to-peer network is automatically established between the RCE instances running on different machines, making components visible and executable even between instances that are only connected indirectly. The distributed execution capability alleviates tool deployment issues, Figure 131, including those related to the protection of intellectual property.

RCE supplies a graphical editor for creation of workflows, using the built-in components to control the data flow. Some built-in components can be used to perform optimization tasks within the workflow, including nested loops, using built-in or user integrated optimization algorithms. After integrating the tools required for the execution of the workflow, the user may compose them into a workflow. To this end, RCE offers a graphical editor allowing the user to construct a workflow by first dragging and dropping the required components into the editor and subsequently connecting their respective inputs and outputs. After constructing such a workflow, the user can execute it. The data model CPACS has been introduced and developed at the German Aerospace Center (DLR) since 2005. CPACS

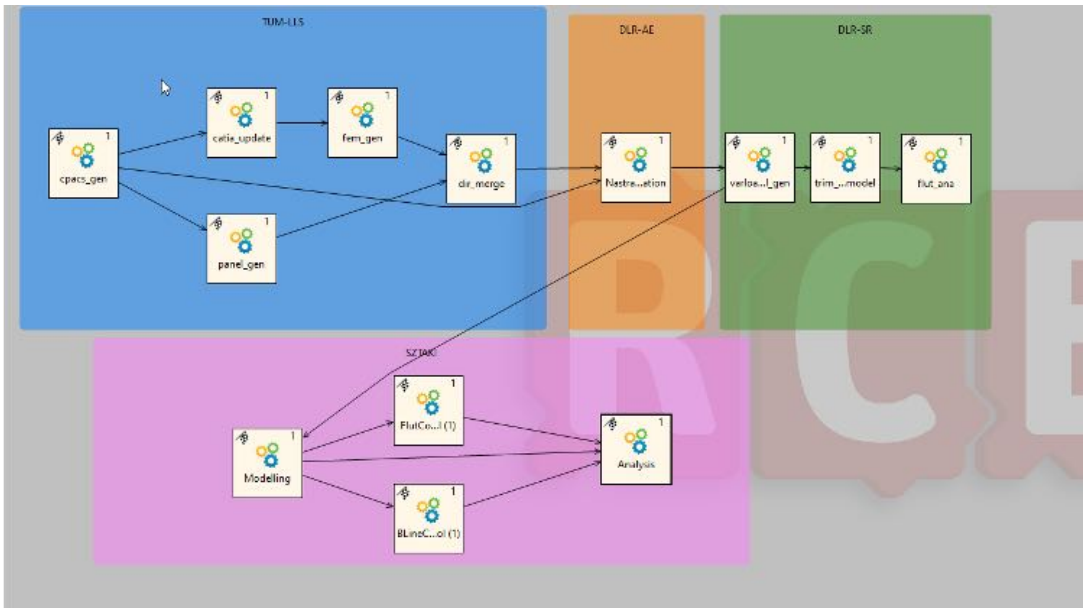


Figure 132: Implemented RCE workflow

is implemented in XML. The data of the aircraft and the resulting controllers are stored and shared between the blocks via CPACS.

The final implementation of the workflow can be seen in Figure 132. The figure also indicates the responsible partners.

Control oriented modelling of the T-Flex demonstrator aircraft

The MDO toolchain includes several control design blocks. These block require a control oriented model of the aircraft, which serves as the foundation for the control design. In the following, the key aspects of the control oriented model development will be presented. The starting point of the aircraft model is the reference Flexop model. This aircraft has the following main characteristics. The aircraft has a wingspan of 7 m and aspect ratio of 20. The aircraft has a 300 N jet engine. The empennage is configured as a V-tail and each wing has 4 control surfaces, [91]. The outer control surfaces are used for flutter suppression, see Figure 133.

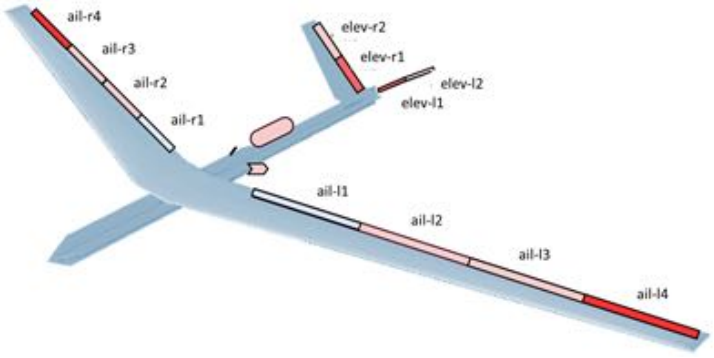


Figure 133: T-Flex aircraft control surface configuration

The reference aircraft has two unstable aeroelastic modes. The first aeroelastic mode (symmetric) goes unstable at 52 m/s and 50.2 rad/s and the second (asymmetric) at 55 m/s and 45.8 rad/s. In order to have sufficient bandwidth, custom made actuators are designed for the aircraft. In addition to the GPS and air data probe, the aircraft has inertial measurement units (IMUs) at the center of gravity and in the wings as shown in Figure 134.

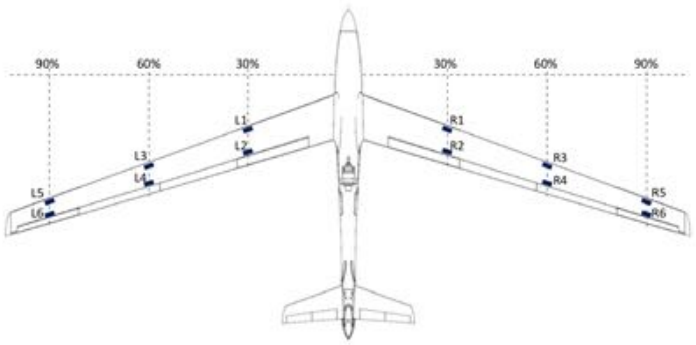


Figure 134: T-Flex aircraft sensor configuration

It is assumed that in the MDO process, the flutter tuning mass, the sweep angle and the laminate directions in the wing can change. This can lead to slightly different flutter modes, flexible behaviour and aerodynamic parameters compared to the reference model. The automatic control oriented model generation needs to account for and to capture these changes in a robust and automatic manner. In the remainder of the section, the main aspects of the model generation are presented in this respect.

Dynamical high complexity model construction for baseline, MLA, GLA and flutter (TUM)

First, a high fidelity nonlinear model of the aircraft is constructed. From this high fidelity model the control oriented models for the various control design applications can be derived. Catia model of wing -1 is reconstructed with parametrized platform and structural layout. To increase the number of flaps, three different configuration sets of flaps (4 flaps, 8 flaps and 16 flaps) were modelled in Catia. The current CAD model is fully capable of handling the design parameters. The geometrical modelling process is automated with the Catia macro language. Structure of wing -1 was modelled in HyperMesh (Figure 135.). All the model generation operation was programmed with HyperMesh native macro language TCL. Currently the geometrical and structural modelling tools are integrated in the RCE framework and works automatically without human intervention.

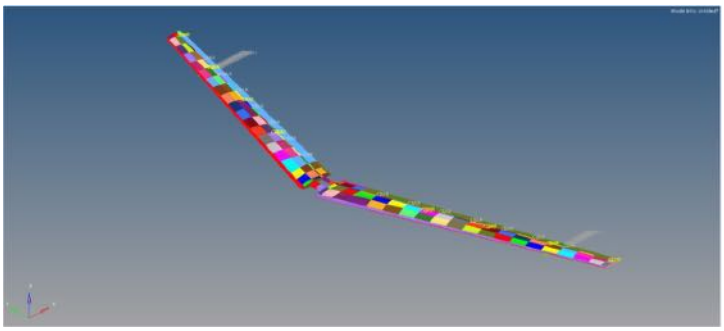


Figure 135: Wing FEM generation

The aerodynamic modelling tool for aeroelastic analysis is developed with PyNastran. It is implemented with a CPACS interface to ease the data input and has a default Nastran output. The tool is also

integrated into RCE.

The aerodynamic solver for drag estimation would be finally integrated into MDO toolchain, which make a high request for the calculation speed (Figure 136).

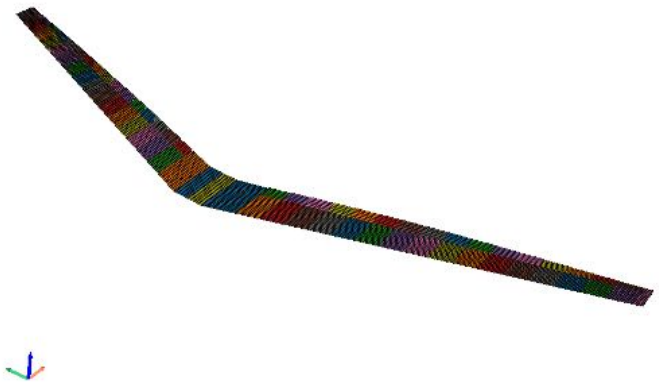


Figure 136: Wing DLM generation

So several VLM-based aerodynamic tools are investigated for instance AVL, PyTornado and VSPAERO. And CFD simulations are carried out with StarCCM+ and SU2 to provide the baseline for comparison. Up to writing the final decision of aerodynamic solver is not made. The comparison of different aerodynamic solver is still in progress. This model serves as a baseline for what follows. The TUM workflow is shown in Figure 137.

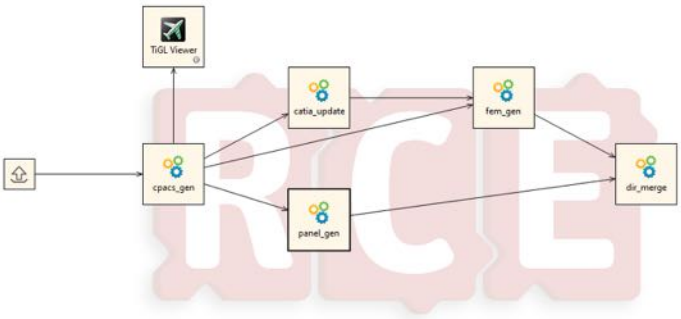


Figure 137: TUM workflow

Control oriented model construction for MLA (ONERA)

Considering the two main issues to automatise the AC system design process within the context of Manoeuvre Load Alleviation (MLA) control system, we have tried to develop a simple and intelligible process based on established methods and our knowledge from the practical constraints of the aeronautical industry. Alternative approaches are clearly possible. Besides, the focus has been placed on robustness of the process more than the performance of the solution. The resulting control-law is

therefore not expected to perform as well as a carefully hand-tuned one.

The retained architecture for the MLA module is schematised in figure 138. For sake of simplicity and ease of interconnection in the MDO tool-chain, its inputs are reduced to the bare minimum:

- the initial large-scale aero-servo-elastic (ASE) model coming from the physical modelling of the aircraft. Note that what *large-scale* means largely depends on the considered domain. For AC, a few hundred of states is already considered as large-scale and prevents from exploiting modern synthesis or analysis tools which can be numerically demanding. This seldom explains the need for the reduction sub-module.
- The specifications for the MLA control-law are simply the target response time for the tracking and the sought complexity of the controller. Filters and other tuning parameters specific to the retained synthesis framework are kept internal to the module.

In output, the module returns

- the control-law K given by its state-space realisation,
- a performance indicator indicating whether the objective response time and dimension of the control-law are achieved.

The module itself is functionally divided in three blocks:

- model reduction: reduce the number of state of the ASE model to a tractable number,
- control synthesis: find a stabilising and structured control-law to ensure tracking while minimising the loads,
- analysis: determine whether the resulting control-law achieves the global tracking objective on the large-scale model.

Model reduction

In order to reach the simplicity and robustness objective mentioned in the introduction, only well established reduction frameworks have been considered [2, 3]. Those frameworks are generally restricted to Linear Time Invariant (LTI) models and stand as follows:

Considering a stable large-scale LTI model H or dimension n, find a reduced-order model H_r of dimension r so that $\|H - H_r\|$ is small in some sense.

This readily sketches some limitations:

- In practice, most methods assume that the reduction order r is given[3]. Yet it is not a very meaningful decision parameter as different models may require very different complexity to be represented accurately.
- Usual systems norms considered to evaluate the error ($\mathcal{H}_2$-norm or more rarely $\mathcal{H}_\infty$-norm) do not necessarily translate easily some practical constraints such as: preserving static gain, preserve equally all channels, etc. Part of these issues can be address by proper filtering which add more tuning parameters.

[3]To mitigate this assertion:*(i)* Balanced Truncation (BT) and Loewner Framework (LF) require rather a tolerance on some singular values which translates into an order and *(ii)* with the Optimal Hankel Truncation, the given order is a maximum value, which is already better. Unfortunately, the method has not proven very efficient in practice

Figure 138: Overall scheme of the automated MLA process

- The initial model is generally assumed to be stable, and most method also needs it to be well-conditioned. This may not be the case and several spurious low frequency poles are common with ASE models.
- As the open-loop error is considered, a low error does not necessarily translate in similar closed-loop behaviour with some controller K.

First point is addressed by a dedicated order selection approach, the second point by implementing a framework that has proven suitable for past aeronautical applications and the third point is dealt with a pre-treatment process. The last point can be dealt with using robust control techniques (see e.g. [126]) yet this generally induces some conservatism and is not exploited here. Instead, an a posteriori validation of the law on the large-scale model is preferred and usually performed.

Two reduction techniques have been considered in the module: the Balanced Truncation (BT) and the Loewner Framework (LF). These methods are thoroughly described in the literature (see e.g. [2] and [3]) and only the basic ideas are recalled:

- BT consists in truncating the state-space representation in the balanced basis so that only the most observable and controllable states are kept. In addition to preserve the stability of the large-scale model, the $\mathcal{H}_\infty$ approximation error is bounded by twice the sum of the discarded singular values σ_i of H, i.e.
$$e_\infty(r) = \|H - H_r\|_\infty \leq 2\sum_{r+1}^{n} \sigma_i = \overline{e}_\infty(r). \tag{30}$$
- From a set of SISO[4] frequency-domain data, the LF enables to build a m-th order descriptor model G_m that interpolates the initial data. Provided that there is enough data and under some rank assumptions involving the Loewner matrix $\mathbb{L}$ and shifted Loewner matrix $\mathbb{L}_s$, the realisation can be projected to an order $k \leq m$ without affecting the interpolation. The resulting model G_k is minimal with a McMillan degree given by $rank(\mathbb{L})$. In practice, this rank is computed numerically and thus involves some tolerance.

Both frameworks are thus quite different but the next sections show how they complement each other to form the main elements of the MLA module.

Pre-processing of the model

In addition to their dimension, three issues are generally encountered with ASE models: instability, presence of delays and difference in magnitude of the inputs and outputs. Each point is described below with an adequate counter-measure.

Instability: ASE models may embed low frequency poles which are either marginally stable or unstable. These either correspond to rigid-body dynamics of to numerical artifacts. For the MLA case and following the cascaded control architecture, true unstable dynamics should be taken care by other control modules and they should thus not be modified by the MLA. Therefore, the ASE model for the MLA synthesis should be expected to be stable. Unstable components should thus be discarded.

Stable/unstable decomposition of finite dimensional LTI models is available in Matlab. It is performed prior to the embedding of delays into the ASE model.

[4] In the MIMO case, the interpolation is fulfilled along some prescribed tangential directions.

Delays: Due to the modelling of the aerodynamic forces on the fuselage or due to the embedding of other control systems loops, input ASE models can contain input/output or internal delays.

While time-delay systems have been widely studied in the literature, associated reduction or control methods remain complex and difficult to implement in an automated manner. Therefore, a practical and standard approach is used instead. It consists in approximating delays through rational functions to fall back on a finite dimensional LTI system. In practice, this approximation step is usually handled through Padé approximation which is available in Matlab. Here: *(i)* Padé is indeed used prior to the BT and *(ii)* as the LF works with frequency-domain data, it is directly fed with the data embedding the delays.

Input-output magnitudes The inputs and outputs gather quantities which magnitudes are largely different (e.g. speed, angle, etc.). In order to preserve equally well all the transfer of the model during reduction, it is necessary to add input/output weighting matrices so that the norm of each channel is comparable.

This is done in the MLA module by adding diagonal scaling matrices in input and in output of the model. The weights are selected to normalise first the norm of each row (i.e. output) and then each column (input) considering either the 2 or ∞ norm. While it is not possible to normalise perfectly all the channels through this process, it significantly decreases the discrepancies that can appear between various channels during reduction thus achieving a better matching from a practical point of view.

Note that this normalisation process is also very useful for the synthesis process as it eases the selection of weighting functions.

Automatic order selection

The most straightforward approach comes from the BT technique which offers an interesting upper bound on the approximation error through the Hankel Singular Values (HSV). Note that the LF directly comes with an estimation of the adequate order to interpolate some given frequency-domain data. Still, it sometimes requires further reduction and what follows can thus be exploited in combination. In particular, the LF and the BT criterion are used jointly in the data-driven approach sketched below.

Dense delay-free case. Due to the bound (31), fast-decaying HSV is generally considered as a relevant indicator to assess the potential for reducing some LTI model. Note that based on Proposition 8.3 of [2], the upper bound (30) can be completed by the following lower bound,

$$\underline{e}_\infty(r) = \sigma_{r+1} \leq e_\infty(r). \tag{31}$$

The bounds (30) and (31) readily suggest a pessimistic or optimistic approach to select the adequate approximation order. Indeed, considering some target relative error e:

- if the $\mathcal{H}_\infty$-norm $N_\infty = \|H\|_\infty$ of the large-scale model is available, one can seek for the order r such that
$$\underline{re}_\infty(r) = \frac{\underline{e}_\infty(r)}{N_\infty} \leq e \quad \text{or} \quad \overline{re}_\infty(r) = \frac{\overline{e}_\infty(r)}{N_\infty} \leq e. \tag{32}$$
- if N_∞ is not available due to the dimension of the model, then one can combines the bounds (30) and (31) with
$$\sigma_1 \leq \|H\|_\infty \leq \|\sigma\|_1, \tag{33}$$
to obtain the following relation for the relative error re_∞
$$\underline{\underline{re}}_\infty(r) = \frac{\underline{\underline{e}}_\infty(r)}{\|\sigma\|_1} \leq re_\infty \leq \frac{\overline{e}_\infty(r)}{\sigma_1} = \overline{\overline{re}}_\infty(r). \tag{34}$$

Obviously, (34) has an increased conservatism in comparison to (32) but it is simpler to compute as it does not involve N_∞.

To highlight the conservatism of these bounds, let us consider the following set $\mathcal{M}$ of test models from COMPLieb [61]: LAH, CDP, DLR2, DLR3, ISS1, CM3 and CM4 (the model TL has been discarded, see remark 1). They have been selected for their resonant nature, a characteristic that is often shared by aeroelastic models. These models are reduced with BT for various orders ranging from 1 to $min(n/2, 50)$ and the relative error re_∞ is computed together with the various bounds (32) and (34). The ratios of the upper bound with the true relative error is reported in figure 139 and the ratio of the relative error with the lower bound is reported in Figure 140.

One can see that the conservatism of the upper bound increases with the approximation order while the lower bound has a more constant conservatism. As expected, using N_∞ in (32) is more accurate (blue dots) than using its bounds in (34) which increases even more the conservatism (red dots). Again, the effect is more visible on the upper bound which is on average 8 times larger than the true error when using N_∞ and 15 times larger when it is not exploited. With the lower bounds, the mean values are 2 and 3, respectively.

Remark 1 (Numerical issues associated with the model TL)**.** *Despite its resonant nature, the model TL has been discarded of the results as it led to various numerical issues. In particular, for approximation orders larger than* 35*, both the upper bounds and lower bounds were invalid. For* $r = 35$*, the reduced-order model is already extremely accurate. This illustrates that choosing an unnecessary large order* r *can be counter-productive as it renders the reduction process numerically more sensitive.*

This also shows that the bounds (32) *and* (34) *should be considered with care in practice as they involve quantities (the HSV) that may be numerically sensitive (see [2, chap. 7]).*

Let us now consider the 393-th order MLA model from Flipased *H* which has been pre-processed. The model is reduced for various orders ranging from 1 to 40 with BT and the approximation error is computed together with the different bounds presented above. The results are reported on figure 141 and are coherent with the previous observations. In particular, avoiding N_∞ increases the conservatism and the lower bounds are closer than the upper bounds to the true error.

Suppose that a relative approximation error $e = 5\%$ is sought. This level of accuracy is reached with a 4-th order reduced model. This order is also suggested by both lower bounds while the upper bounds $\overline{re}_\infty$ and $\overline{\overline{re}}_\infty$ suggest 12 and 15, respectively.

All in all, these tests show that the HSV can provide meaningful information on the adequate approximation order r to reach some prescribed level of accuracy. Optimistic or pessimistic estimations are given through the upper/lower bounds (32) and (34). These bounds can be combined to derive a mixed criterion, e.g.

$$(1-\alpha)\underline{re}_\infty(r) + \alpha\overline{re}_\infty(r), \tag{35}$$

where $\alpha \in [0, 1]$ is a tuning parameter to adjust the compromise of approximation order against the requirement that the resulting order enables to reach a prescribed approximation error.

Still, this approach implies a significant numerical burden as the HSV are obtained at the cost of solving two Lyapunov equations. The approach is thus only suited for dense models of moderate size. In addition, it does not handle delays which must be dealt with separately by ad-hoc methods (e.g. with Padé as above).

Data-driven case. A recent article [29] investigates the use of input-output data to approximate the gramians for use in BT. More specifically, it is shown how the evaluation of the transfer function can

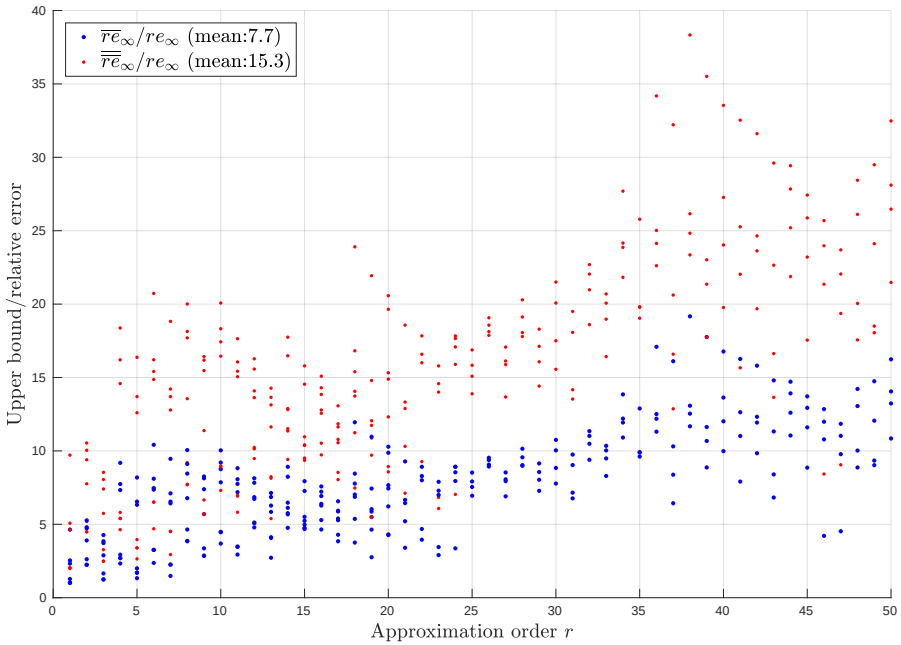


Figure 139: Ratio between the upper bounds of the approximation error and the actual error for various orders and models in $\mathcal{M}$

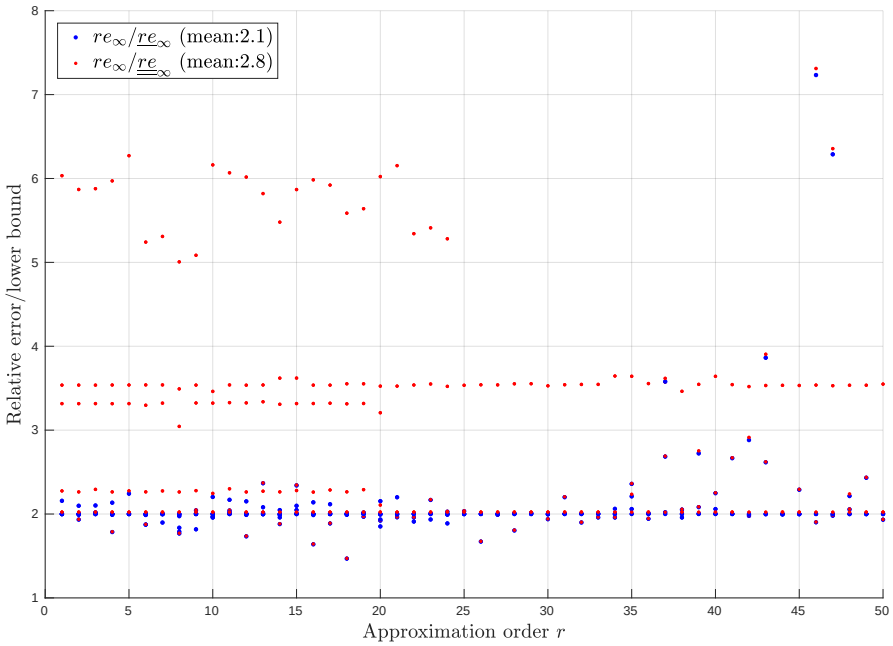


Figure 140: Ratio between the actual approximation error and lower bounds for various orders and models in $\mathcal{M}$

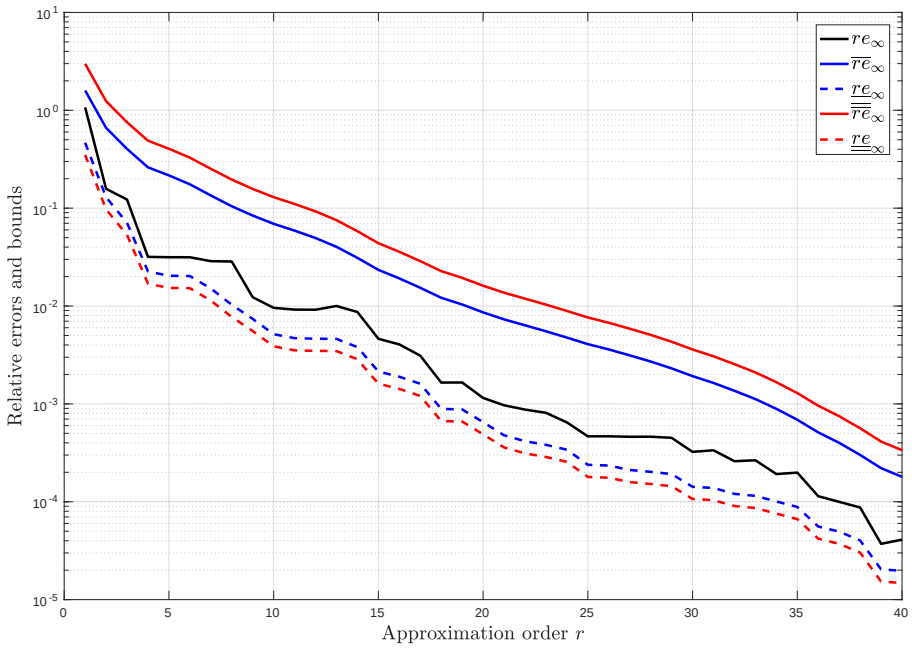


Figure 141: Approximation errors and associated bounds for the MLA model of Flipased

be exploited to approximate the integral definition of the gramians. The approach is also linked with a Loewner-based approach. In this context, various ideas stem for reduction purpose

- use the minimal model G_k from the LF to compute the gramians as a surrogate for H to compute the HSV and apply the approach detailed in the previous section. In the sequel, this is referred to as Loewner-HSV. Note that only stable HSV are considered as the unstable part of the model to be reduced needs to be kept anyway.
- Reason directly on the singular values of $\mathbb{L}$ (or the pencil $(\mathbb{L}, \mathbb{L}_s)$). While they are not the HSV of the model, their decay embeds the information about the minimal order k of the interpolating model (through the rank of the matrix) and can therefore also be relevant. In the sequel, this is referred to as Loewner-SV.

The second point is considered in [29] in comparison to their approach. It is illustrated that the singular values of the Loewner matrix follows the trend of the HSV but are not of the same magnitude. The first point on the other hand, has not been evaluated by the authors of [29].

To compare Loewner-HSV with the approach developed in [29], let us consider one of the example the authors give in section 3.4.1. It compares the HSV of the ISS1 model with their estimation. For the data-driven approaches, the authors use a grid of 400 frequency points logarithmically spread from 10^{-1} to 10^2. The resulting singular values are reported in Figure 142.

First, note that the Loewner-SV reach machine precision after $i \sim 170$. This gives the minimal order k for the interpolant model G_k. This is why there are less HSV with Loewner-HSV. The Loewner-SV indicates that there is no additional information that can be extracted from this set of data. And globally, the model G_k is extremely accurate as $\|H - G_k\|_\infty / \|H\|_\infty = 0.07\%$. This is not surprising considering the number $N = 400$ of interpolation points in comparison to the dimension $n = 270$ of the model.

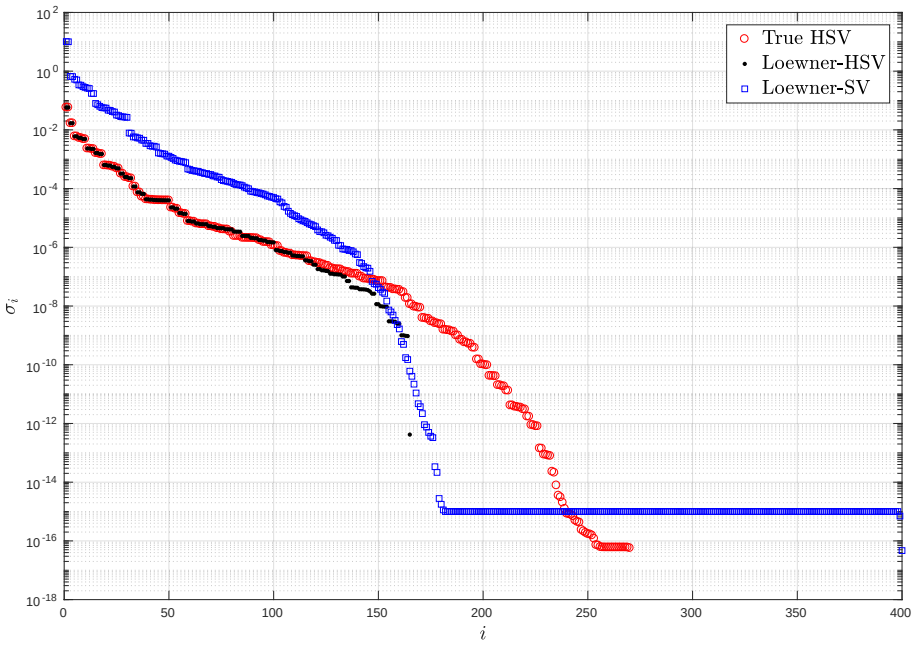


Figure 142: Comparison of the true HSV of the ISS1 model with the Loewner-HSV and the Loewner-SV

In addition, we retrieve the results from [29] and we can observe the same scaling factor between the true HSV (red circles) and the Loewner-SV (blue squares). On the other hands, the Loewner-HSV (black dots) appear to be extremely accurate (up to k) and comparable to the ones obtained with the dedicated method in [29].

Illustration of the reduction process

The reduction process is applied to the flipased ASE model with 426 states and 3 inputs, 4 outputs, 9 internal delays and 3 output delays. Its frequency response is displayed together with the ones of the reduced-order models obtained with BT and LF process.

The corresponding models H_{71}^{BT} and H_{35}^{LF} have not the same dimension. This comes from the way delays are handled and the tolerance on the selection of the approximation order which is based on the HSV for the BT and on the Loewner-SV for the LF. Still, both models are representative of the main dynamics of the initial model up to the prescribed frequency of interest (shaded area). Note that thanks to the normalisation process, the lower singular values are still matched.

Dynamical high complexity model construction for GLA (DLR)

The gust input for the loads simulation is a discrete, vertical 1-cosine function. It is given by the gust zone velocity

$$U_{z,\text{gust}}(t) = \begin{cases} \dfrac{U_{\text{gust,max}}}{2}\left(1 - \cos\left(\dfrac{\pi}{H_{\text{gust}}} U_\infty t\right)\right), & \text{if } t_{z,1} \leq t \leq t_{z,\text{end}} \\ 0, & \text{otherwise,} \end{cases} \tag{36}$$

where $U_{\text{gust,max}}$ depicts the maximum gust velocity, H_{gust} the gust half length [24] and U_∞ the airspeed. With passing time t the aircraft moves through the gust, as shown in Figure 144 for the demonstrator

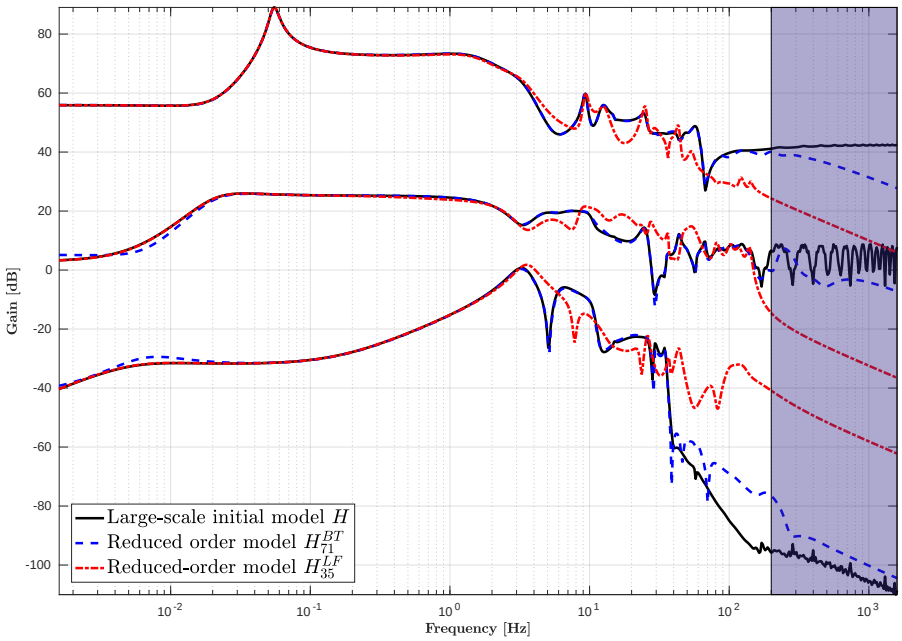


Figure 143: Reduction of the delayed MLA model with the BT and LF approaches

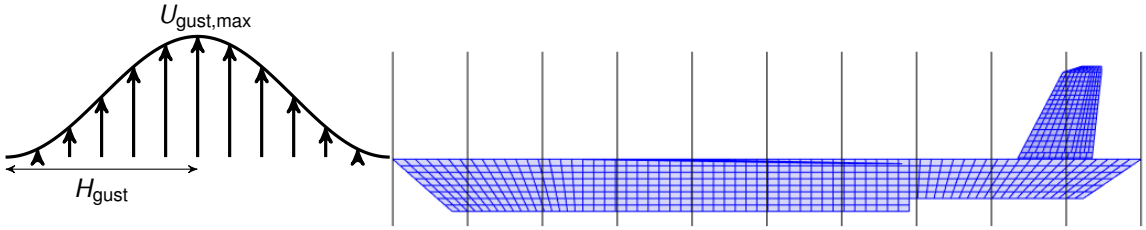


Figure 144: 1-cosine gust and aircraft gust zones

aircraft. At time $t_{z,1}$ the gust is at the centre of a gust zone and leaves it at time $t_{z,\text{end}}$. The aerodynamic panels (small blue quadrangles) within the same gust zone experience the same gust velocity, to save computation time [44]. The difference in the gust zone velocity of two neighbouring gust zones is simply a time delay, which has a transfer function of the form

$$G_{z,\text{delay}}(s) = e^{-t_{z,\text{delay}}s}, \tag{37}$$

with $t_{z,\text{delay}}$ being the time delay in seconds and s the Laplace variable [44]. Equation (37) is approximated by a second-order Padé filter

$$G_{z,\text{delay}}(s) \approx \frac{s^2 - \frac{6}{t_{z,\text{delay}}}s + \frac{12}{t_{z,\text{delay}}^2}}{s^2 + \frac{6}{t_{z,\text{delay}}}s + \frac{12}{t_{z,\text{delay}}^2}}. \tag{38}$$

in order to make it convertible to a state-space format [34].

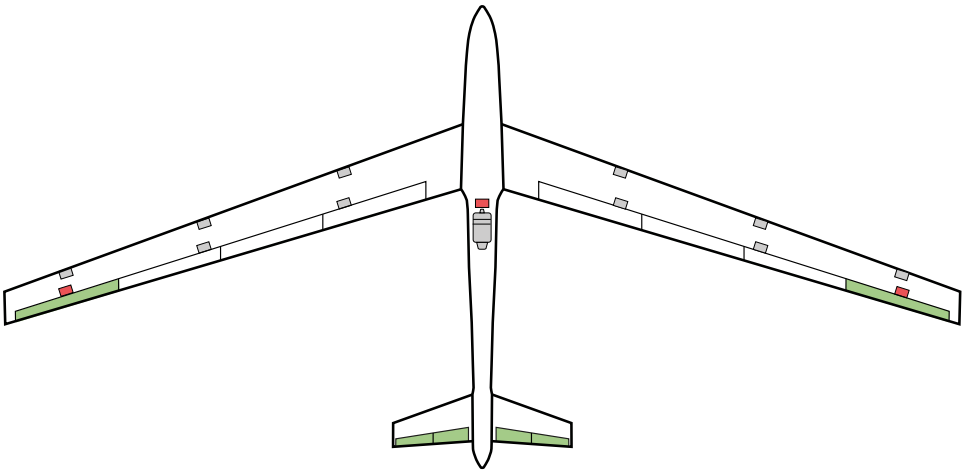


Figure 145: Demonstrator aircraft with IMUs (red) and control surfaces (green) for GLA control [66]

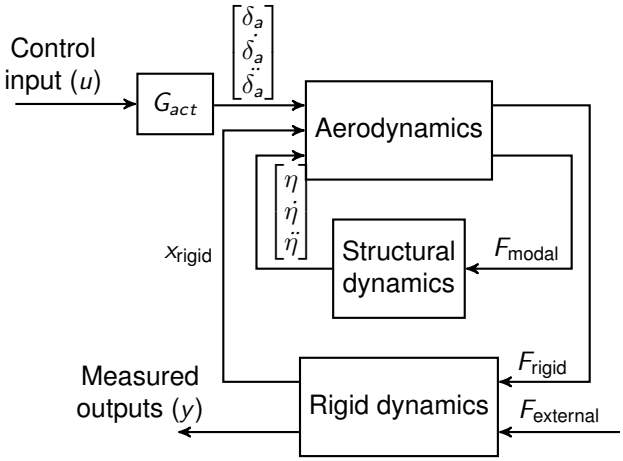


Figure 146: ASE subsystem interconnection

The control surfaces chosen to react to the gust encounter are the most outer ailerons on both wings and all elevator control surfaces, like for the demonstrator aircraft in Figure 145. The inputs to these control surfaces are allocated by utilising the longitudinal symmetry of aircraft. This leads to two inputs u_{ail} and u_{ele}, one signal u_{ail} connected to both ailerons and signal u_{ele} connected to the elevator control surfaces.

Model development for baseline and flutter suppression control design (SZTAKI)

The controllers for the T-Flex demonstrator are designed based on a suitable model. Such model is called the control oriented model and the reminder of this section will describe the main steps of the automated control oriented model development.

Flexible aircraft are typically modelled using subsystems. The structural dynamics model, the aerodynamics model and the flight mechanics model are combined to form the aeroservoelastic (ASE) model. Such subsystem interconnection is depicted in Figure 146. These ASE models in general are of too high order for control design, therefore, model order reduction is required. One approach applied for the MDO process is the bottom-up modeling approach, [107, 70, 118].

The key idea of the bottom-up modeling is the following. The subsystems of the ASE model in general have simpler structure than the nonlinear ASE model. Therefore, the subsystems containing the structural dynamics and aerodynamics model can be reduced by simpler, more tractable reduction techniques. Combining these reduced order subsystems results in a low order nonlinear ASE model upon which a nominal, low order, control oriented models can be obtained. The control oriented models are based on the LPV framework, [95, 7]. The LPV framework can serve as a good approach to model ASE systems for control design. The benefits of utilizing the LPV framework are the following; it can capture the parameter varying dynamics of the aircraft and many of the linear time-invariant (LTI) control design techniques have been extended to LPV systems. An LPV system is described by the state-space model [117, 95]

$$\dot{x}(t) = A(\rho(t))\ x(t) + B(\rho(t))\ u(t) \tag{39a}$$
$$y(t) = C(\rho(t))\ x(t) + D(\rho(t))\ u(t) \tag{39b}$$

with the continuous matrix functions $A\colon \mathcal{P} \to \mathbb{R}^{n_x \times n_x}$, $B\colon \mathcal{P} \to \mathbb{R}^{n_x \times n_u}$, $C\colon \mathcal{P} \to \mathbb{R}^{n_y \times n_x}$, $D\colon \mathcal{P} \to \mathbb{R}^{n_y \times n_u}$, the state $x\colon \mathbb{R} \to \mathbb{R}^{n_x}$, output $y\colon \mathbb{R} \to \mathbb{R}^{n_y}$ input $u\colon \mathbb{R} \to \mathbb{R}^{n_u}$, and a time-varying scheduling signal $\rho\colon \mathbb{R} \to \mathcal{P}$, where $\mathcal{P}$ is a compact subset of $\mathbb{R}^N$. The system is called quasi LPV model if the parameter vector ρ includes elements of the state vector x. The system matrix $S(\rho(t))$ is defined as

$$S(\rho(t)) = \begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} \tag{40}$$

In a grid-based LPV representation ([117]), the system is described as a collection of LTI models $(A_k,\ B_k,\ C_k,\ D_k) = (A(\rho_k)\,, B(\rho_k)\,, C(\rho_k)\,, D(\rho_k))$ obtained from evaluating the LPV model at a finite number of parameter values $\{\rho_k\}_1^{n_{\text{grid}}} = \mathcal{P}_{\text{grid}} \subset \mathcal{P}$. Using the bottom-up modeling approach, a full order and a low order LPV model is obtained. The full order model is constructed without reducing the subsystems, while the low order model is obtained by reducing the structural dynamics and aerodynamics subsystems. The main measure of the accuracy of the low order model is the ν-gap metric, [116]. The ν-gap between the full and low order models is calculated frequency wise at each grid point.

Modeling block inputs

The modeling block in RCE takes the structural dynamics (M_{hh}, K_{hh}, B_{hh}) and aerodynamics data (Q_{hh}) as input via CPACS. These parameters are expected to change due to the MDO optimization, the rest of the model properties are assumed to be fixed.

Reduction of the structural dynamics model

The structural dynamics of the aircraft are of the form

$$\mathcal{M}\ddot{\eta} + \mathcal{C}\dot{\eta} + \mathcal{K}\eta = F_{\text{modal}} \tag{41}$$

where F_{modal} is the force acting on the structure in modal coordinates, $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{K}$ are the modal mass, damping and stiffness matrices respectively. The full order structural model contains 50 modes, thus the structural dynamics model consists of 100 states. The structural dynamics model is an LTI system, thus state truncation can be applied.

Reduction of the aerodynamics model

The aerodynamic lag terms take the state-space form

$$\begin{aligned} \dot{x}_{\text{aero}} &= \frac{2V_{\text{TAS}}}{\bar{c}} A_{\text{lag}} x_{\text{aero}} + B_{\text{lag}} \begin{bmatrix} \dot{x}_{\text{rigid}} \\ \dot{\eta} \\ \dot{\delta}_{\text{cs}} \end{bmatrix} \\ y_{\text{aero}} &= C_{\text{lag}} x_{\text{aero}} \end{aligned} \tag{42}$$

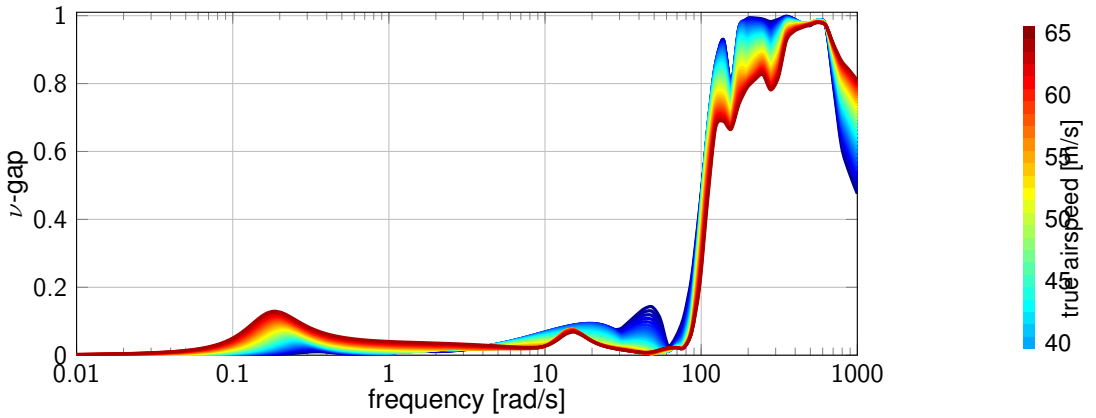


Figure 147: ν-gap values between the nominal low order and high-fidelity models.

where V_{TAS} is the true airspeed, x_{rigid} is the rigid body state, η is the modal state of the structural dynamics, δ_{cs} is the control surface deflection and $\bar{c}$ is the reference chord. The full order model consists of 1040 lag states. Using the aerodynamics model given by A_{lag}, B_{lag} and C_{lag} in (42) an LTI balancing transformation matrix T_{b} is computed. The balanced states of the aerodynamic model with the smallest Hankel singular values are residualized, leading to a reduced order aerodynamics model.

The initial model order reduction produced the following results. The structural dynamics model can be reduced in the following way. In order to keep the ν-gap between the high fidelity and the low order model low the first six structural modes are retained for the reference aircraft model. Te removal of the latter results in a large increase in the ν-gap. This way, a 12 state structural dynamics model can be obtained from the 100$^{\text{th}}$ order model. In case of the aerodynamics model, retaining two lag states results in a low order model with acceptable accuracy. The resulting nonlinear ASE bottom-up model has 26 states that consists of 12 rigid body states, 12 structural dynamics states, 2 aerodynamic lag states. Note, that the actuator dynamics are not included in the control oriented model. The ν-gap between the full order and the reduced order model of the reference aircraft model for different airspeed values is given in Figure 147.

Uncertain low order model

The next step is to develop uncertain LPV models of the aircraft. Uncertain models can be developed by extending the structural dynamics model with the uncertain parameters. These uncertainties appear in the mass matrix $\mathcal{K}$ and in the damping matrix $\mathcal{C}$ in (41) of the nonlinear ASE model and are denoted by $\delta_{\mathcal{K}}$ and $\delta_{\mathcal{C}}$, respectively. Based on this uncertain, nonlinear model a grid-based uncertain LPV model is constructed. The grid-based uncertain LPV model is obtained over a 3 dimensional grid. The grid consists of 81 equidistant points of the airspeed between 30m/s and 70m/s, 3 points of the natural frequency in the structural dynamics between $\pm 1\%$ of the nominal value, and 3 points of the damping in the structural dynamics between $\pm 10\%$ of the nominal value. This results in a total of $81 \times 3 \times 3 = 729$ grid points. The scheduling parameter ρ can then be defined as

$$\rho = \begin{bmatrix} \rho_{V_{TAS}} \\ \delta_{\mathcal{K}} \\ \delta_{\mathcal{C}} \end{bmatrix} \tag{43}$$

where $\rho_{V_{TAS}}$ is a measured parameter and $\delta_{\mathcal{K}}$ and $\delta_{\mathcal{C}}$ are unmeasured. These uncertainties have a significant effect on the flutter speeds and frequencies. The nominal and uncertain flutter modes of the control oriented LPV model are shown in Figure 148.

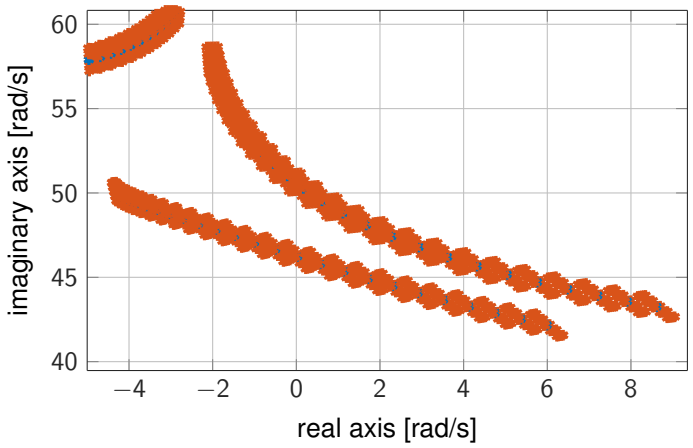


Figure 148: Uncertainty of the flutter modes: nominal model (blue), uncertain (red)

Modeling block robustness

As it can be seen, the bottom-up modeling approach involves a certain degree of heuristics. These heuristic steps include the selection of the structural dynamics states to retain and setting the the number of retained aerodynamic lag states. These parameters are hand tuned for the initial, reference aircraft model. The modeling tool needs to be adopted to the collaborative design in this respect. This means that the retained the initial structural modes to be retained are the ones of the reference aircraft. However, it is crucial that after every MDO iteration, the ν-gap metric is analyzed and that it does not exceed a threshold value. If this value is exceeded, it means that the bottom up-model is not accurate enough. Therefore, at the expense of increasing the order of the resulting model, additional structural modes need to be retained. The number of retained modes is increased until the ν-gap values are satisfactory. A similar approach is used for the order of the lag state aerodynamics model. In this case the number of the retained lag states is increased until a satisfactory ν-gap level is obtained.

Modeling block outputs

The modeling block provides two LPV models, one for the baseline control design (RigACModel) and one for the flexible control design (FlexACModel). The FlexACModel is the low order, uncertain LPV model of the aircraft obtained by the steps described above. The RigACModel is obtained from the nominal low order aircraft model by rezidualizing the structural and lag state dynamics. This model serves for the baseline control design, containing only the 12 rigid body states. These resulting models are saved in the ToolSpecific section of CPACS.

Baseline control design functions (SZTAKI)

The starting point for the baseline control design is the baseline design for the flight tests. However, some modifications had to be carried out. The reason for this is that flight test controllers required a lot of hand tuning to achieve the highest performance. The key idea is to simplify these requirements in order to achieve controller design alrorithms that can run in an automatic fashion.

The baseline control design is based on the LPV model obtained in the model integration block, that has 12 rigid body state and the actuator dynamics. The baseline control design takes the actuator dynamics and the baseline control design model RigACModel as inputs via CPACS. The baseline control system features a classical cascade flight control structure with scheduled control loops to augment the lateral and longitudinal axis of the aircraft, see Figure 149. The control loops use scheduled elements of proportional-integral-derivative (PID) controller structures with additional roll-offs in the inner

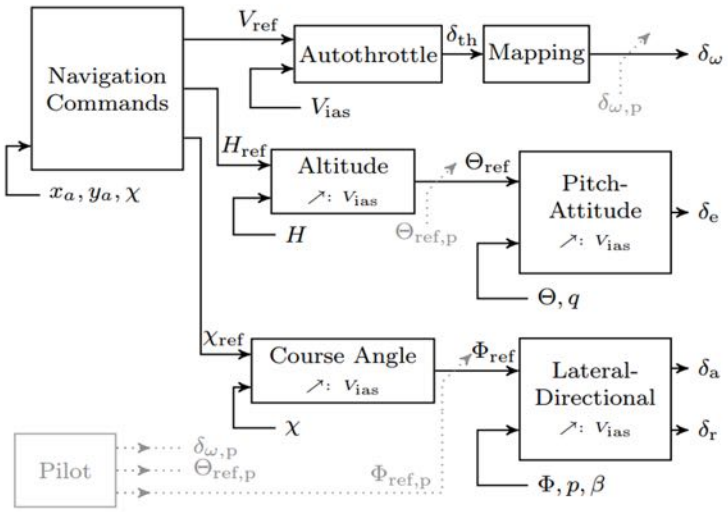


Figure 149: Baseline control architecture

loops to ensure that no aeroelastic mode is excited by the baseline controller. Scheduling with indicated airspeed V_{ias} is used to ensure an adequate performance over the velocity range from 40 m/s to 70 m/s. Structurally the controller consists of several loops targeting different dynamical modes. Accordingly, intuitive design specification for the loops can be formulated by the user in terms of settling times, reference tracking or robustness margins. The control design itself automatically optimizes the corresponding gains, in order to satisfy the specified design goals. Once the optimization found a feasible solution it provides the corresponding control gains and control structure which is then used for the numerical analysis. The main algorithms of the baseline control design based on the approach described in [63]. The tool used to obtain the respective PID gains is `systune` of Matlab, which provides intuitive access to performance specifications and which well suited for implementation as automatic control design algorithm. In addition, several steps are included in order to make the automatic control design algorithms more robust against model variations resulting from the MDO process. These steps are detailed the controller description.

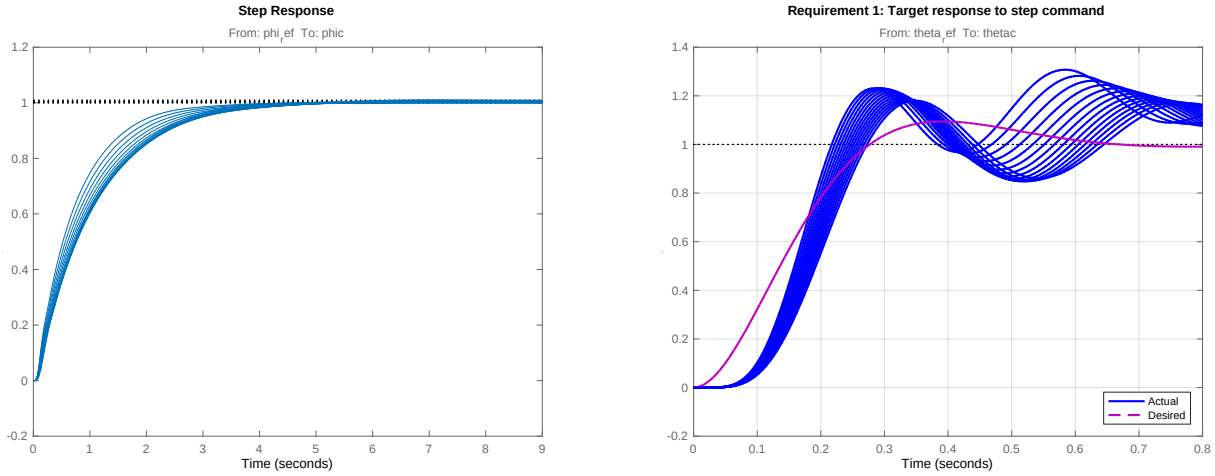
Inner lateral loop design

The inner lateral loop takes the LPV model containing the lateral states, inputs and outputs. Time and frequency domain control performance specifications are given. The time domain performance requirements are formulated as a step reference system tracking. The reference system is given as a second order system

The ideal model is given as

$$G_{id_{lat}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $\omega_n = 0.7 \times \omega_{n_{roll}}$ is the natural frequency and $\zeta = 0.8$ is the damping factor of the ideal model. $\omega_{n_{roll}}$ is the roll frequency of the lateral model. Such approach improves the robustness of the control design approach with the expense of achieving a sub-optimal performance. The frequency domain specifications are $5dB$ gain and $40°$ phase margins. In case the performances cannot be met, the design is repeated with design criteria relaxed by 10% until a feasible controller is found. The time domain performances of the inner lateral control loop can be seen in Figure 150a.



(a) ϕ, lateral loop

(b) θ, longitudinal loop

Figure 150: Time domain performances of the inner loops

Outer lateral loop design

Once a feasible inner controller is designed, the controller is connected to the lateral LPV model and the outer, χ, loop design starts. The outer loop performance is set up as a step tracking requirements. The ideal model is given as

$$G_{id_{lat}} = \frac{0.2}{s + 0.2}$$

Since the model is generally numerically ill conditioned, it can happen that integrators in the model appear as stable or unstable real poles with values in the order of magnitude of $10^{-}9$. In order to overcome such numerical issues, a soft constraint for the outer lateral loop is to have all closed loop poles ≤ -0.01. Similarly as in the inner loop, in case the performances cannot be met, the design is repeated with design criteria relaxed by 10% until a feasible controller is found. The χ loop time and frequency domain performances can be seen in Figure 151.

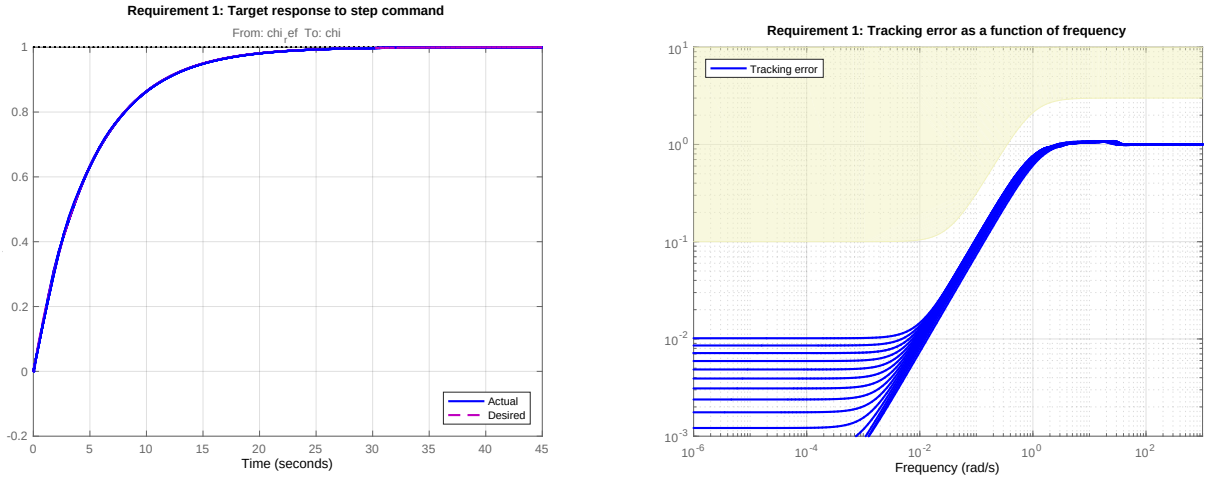
Inner longitudinal loop design

The inner longitudinal loop takes the LPV model containing the longitudinal states, inputs and outputs. The performance is set up as a step tracking requirements. The ideal model is given as

$$G_{id_{long}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $\omega_n = 1/3 \times \omega_{n_{short\ period}}$ is the natural frequency and $\zeta = 0.8$ is the damping factor of the ideal model. $\omega_{n_{short\ period}}$ is the short period frequency of the longitudinal model. In addition to the tracking requirement, $5dB$ gain and 40° phase margins are required as well. If the performances cannot be met, the design is repeated with design criteria relaxed by 10% until a feasible controller is found. The inner longitudinal loop time domain performance can be seen in Figure 150b.

Outer longitudinal loop design



(a) time domain

(b) frequency domain

Figure 151: Time and frequency domain performances of the χ loop

Before the outer, h, loop design is started, the inner loop controller is connected to the longitudinal model. Time domain control performance specifications are given as response time of $3s$, steady state error of 0%, peak error as 1.2. Since the model is generally numerically ill conditioned, it can happen that integrators in the model appear as stable or unstable real poles with values in the order of magnitude of 10^-9. In order to overcome such numerical issues, a soft constraint for the outer longitudinal loop is to have all closed loop poles ≤ -0.01. Similarly, these conditions are automatically relaxed in case a feasible controller is not found. The results of the time domain performance specification is given in Figure 152a.

Sideslip loop design

The sideslip, β, loop takes the LPV model connected with the inner and outer controllers. This ensures that the sideslip controller to be designed does not interfere with the lateral and longitudinal controllers. Since the main goal of this loop is to keep $\beta = 0$, the performance is set up as a disturbance rejection for a frequency range between 0 and $1\ rad/s$ and the damping of the closed loop is set to be $\zeta_{cl} > 0.125$. In addition, $5dB$ gain and $40°$ phase margins are required as well.The β loop time and frequency domain performances can be seen in Figure 152b-d.

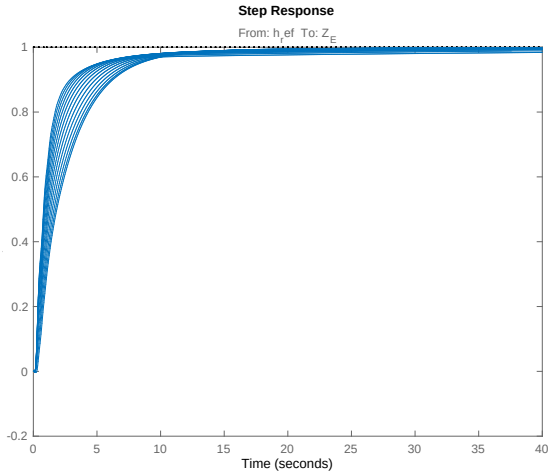
Airspeed loop design

For the airspeed, V_{IAS}, loop the 12 state LPV model is connected with the lateral, longitudinal and sideslip controllers. The performance is set as tracking requirement with response time of $35s$, steady state error of 0, peak error as 1.5. The V_{IAS} loop time and frequency domain performances can be seen in Figure 153.

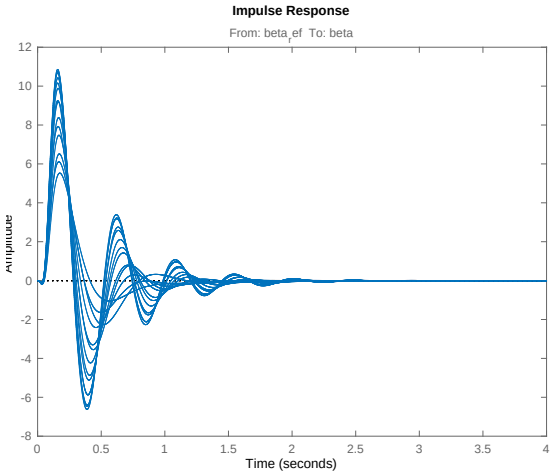
Finally, the 4 controllers response can be seen in Figure 154.

Initial evaluation of the baseline controller

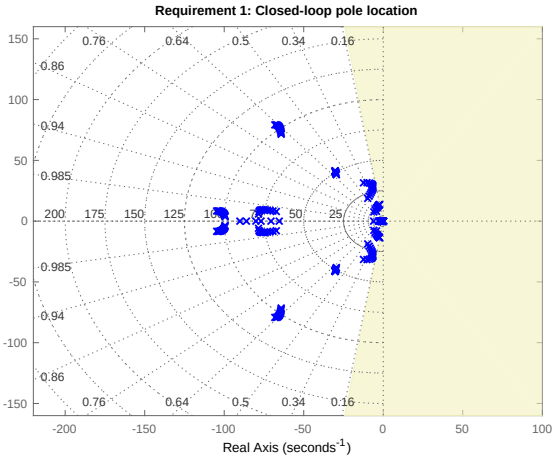
In the first step of the evaluation, it is checked if all of the controllers designed above connected to the 12 state LPV model result in a stable closed loop. In the next step, the controllers are connected to the flexible LPV model (FlexACModel) and verified if the flexible model is stabilized up to the flutter speed. The poles of the baseline controller connected with the flexible model can be seen in Figure 155. It can be seen that the flexible modes are not altered significantly.



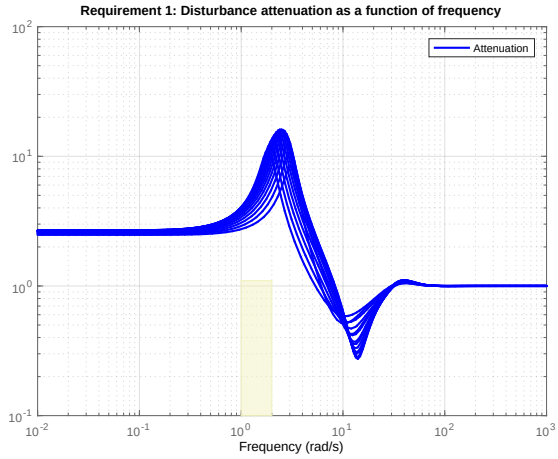
(a) h, time domain



(b) β, time domain



(c) β, frequency domain damping



(d) β, frequency domain rejection

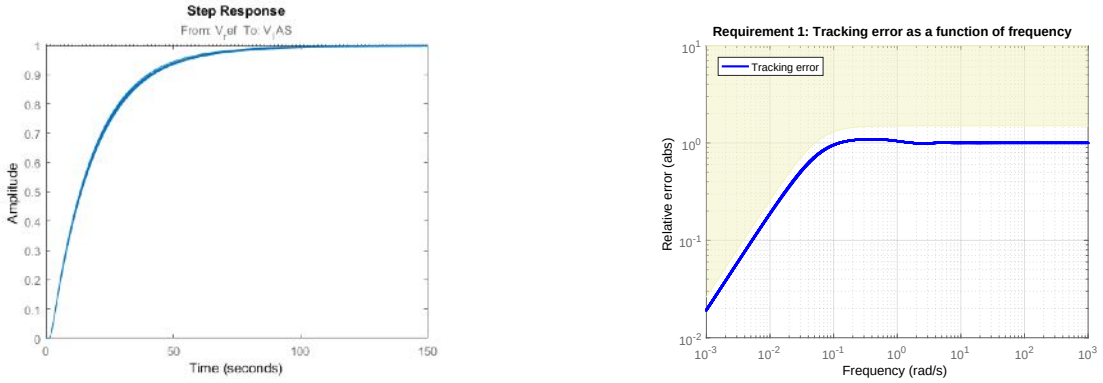
Figure 152: Time and frequency domain performances of the h and β loops

Baseline control design block outputs

The airspeed dependent PID gains for all baseline loops are are saved in the ToolSpecific section of CPACS under the name Baseline. These are the main outputs of the block. A simple metric is also returned for the user which indicates the performance of the control loops. This allows the interaction with the automated design process: the user can formulate tighter or loser specifications according to the individual needs. A clear graphical representation is also provided which can be included in the reporting.

Maneouvre Load Alleviation (MLA) functions (ONERA)

Pre-processing. Now a ROM is available, we are ready to process the control synthesis part. As a preliminary to the MLA optimization, the reduced order model input-output are first normalized. As in the reduction process, this input-output scaling allows dealing equally with all transfer in the optimisation process. Indeed, as detailed in what follows, it allows defining weighting / performance filters in an



(a) time domain

(b) frequency domain

Figure 153: Time and frequency domain performances of the V_{IAS} loop

almost universal manner and unified.

Objective function. Given the obtained reduced order and normalized model, from now on denoted $\hat{H}$, the design of a MLA controller is now possible. We chose the $\mathcal{H}_\infty$ framework for this step. Such a framework aims at attenuating the worst-case of the (closed-loop) transfer $T_{zw}(K) : w \mapsto z$. Such a controller may be obtained using $\mathcal{H}_\infty$-norm oriented tools, e.g., through the solution of the following optimization problem:

$$K = \arg \min_{\bar{K} \in \mathcal{K} \subseteq \mathcal{H}_\infty} ||\mathcal{F}_l(G, \bar{K})||_{\mathcal{H}_\infty} \tag{44}$$

under the stabilizing constraint

$$K \text{ stabilizes } \mathcal{F}_l(G, K).$$

We also denote as γ,

$$\gamma = \min_{\bar{K} \in \mathcal{K} \subseteq \mathcal{H}_\infty} ||\mathcal{F}_l(G, \bar{K})||_{\mathcal{H}_\infty} = ||\mathcal{F}_l(G, K)||_{\mathcal{H}_\infty}. \tag{45}$$

Following (44) and the above notations, $\mathcal{F}_l(G, K)$ represents a lower LFT composed of G, a generalized plant that encompasses the ROM computed in the previous step plus the performance weighting functions, and the controller K. The set $\mathcal{K} \subseteq \mathcal{H}_\infty$ is meant to restrict the search of a controller to a given specific structure (this point is detailed in the following). In addition, w stands as the exogenous inputs while z are the performance output. All the magic of this framework stands in the definition of the weights and in the selection of input-outputs couple w, z. This boils down in defining G so that the problem to be solved is a single objective one.

Input output selection for the control set-up. In order to set up the control scheme, let us define the input and output signals selected:

1. the ailerons at patch #4, u_{a4} (sum right and left)
2. the rudders, u_r (sum right and left)

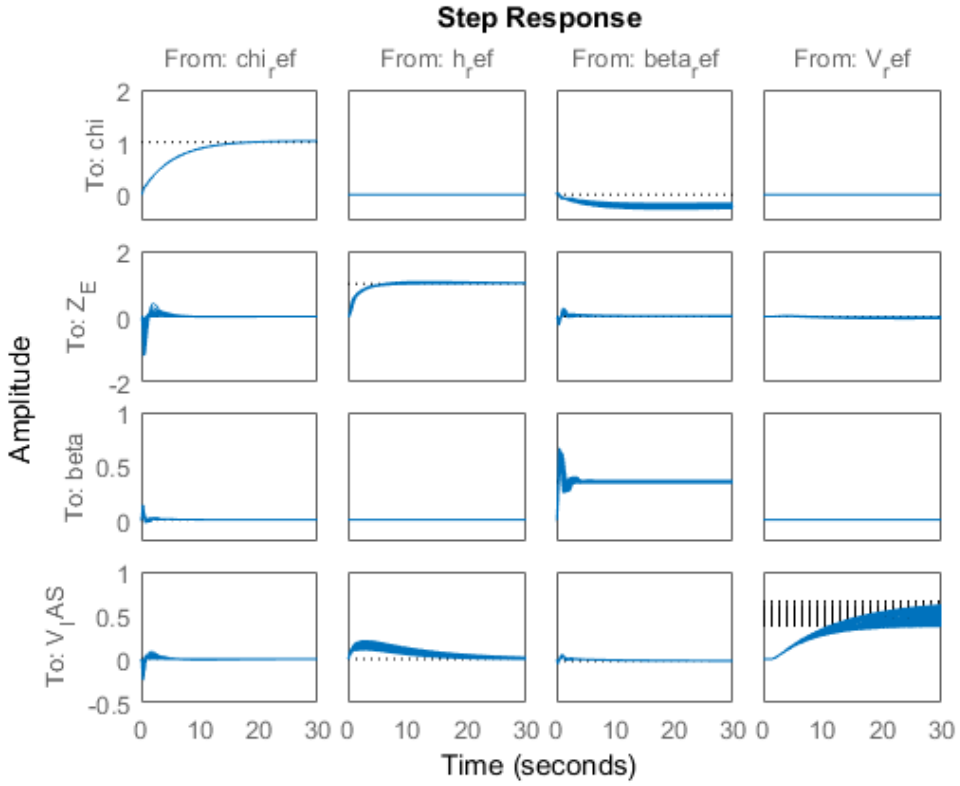


Figure 154: Baseline control response

3. the gust input, w_g (the equivalent gust affecting each longitudinal patch is summed by considering the delay between each patch, function of the velocity and results in a single input)
4. the wing loads, l (computed as loads left plus right divided by two)
5. the pitch angle, θ
6. the pitch rate, q
7. the vertical acceleration, a_z
8. the vertical acceleration reference, $a_z^\star$

Then we denote the following generic signals

$$\begin{array}{rcll} w & = & \text{vec}(a_z^\star, w_g) & \text{(exogenous inputs)} \\ u & = & \text{vec}(u_{a4}, u_r) & \text{(control inputs)} \\ y & = & \text{vec}(\theta, q, a_z^\star - a_z) & \text{(system output measurements for the control)} \\ z & = & \text{vec}(z_1, z_2, z_3) & \text{(performance output, defined hereafter)} \end{array} \tag{46}$$

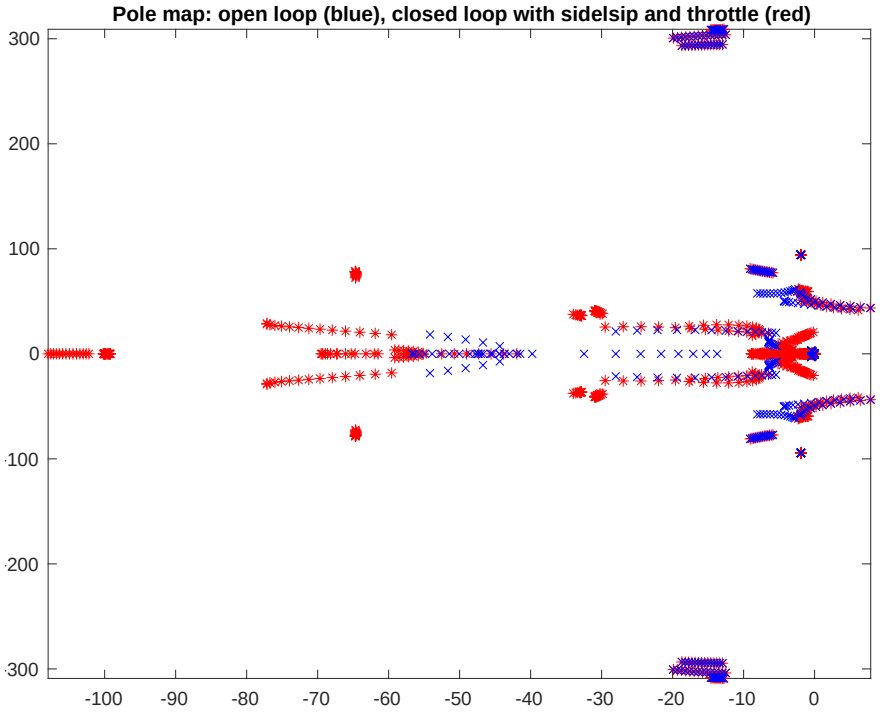


Figure 155: Flexible aircraft dynamics with (red) and without (blue) the baseline controller

Then one defines the plant P as

$$\text{vec}(z, y) = P\text{vec}(w, u) \tag{47}$$

where P is simply a copy of the ROM, considering the input-output rearrangement and scaling (as defined earlier). This latter may be interconnected to the controller K leading to

$$z = \mathcal{F}(P, K)w = P(K)w. \tag{48}$$

We also denote $P_{i\mapsto o}$ the transfer from input i to output o.

Performance definition. The performance are then defined by channels mean of weight in the transfer from w to z (transfer from exogenous inputs to performance outputs). In the MLA they are meant to enforce the following three constraints:

C1- **Pilot load factor tracking error**:

$$z_1 = T_1(K)a_z^\star = W_e(H_r a_z^\star - P_{a_z^\star \mapsto a_z}(K)) \tag{49}$$

where

$$W_e(s) = g_e^{-1}\left(\frac{g_e/\omega_e s + 1}{1/\omega_e s + 1}\right)^2$$

is a weight that allows for ensuring low frequency attenuation (i.e. gain smaller that g_e for frequency below w_e). In addition

$$H_r(s) = \frac{1}{st_r/3 + 1}$$

is an input weight that suggests a tracking reference signal with a response of the form of a first order with response time t_r (seconds). In the considered setup, $t_r = 6$ (response time in 6 seconds), $g_e = 0.1$ (tracking mismatch in low frequency below 10%) and lower limit $\omega_e = 0.1$ rad/s.

C2- **Attenuation of wind to load transfer peaks**:

$$z_2 = T_2(K)w_g = W_p P_{w_g \mapsto l}(K) \tag{50}$$

where

$$W_p(s) = \| T_2(0) \|_\infty^{-1}$$

is the worst-case open-loop gain of $P_{w_g \mapsto l}$. This simple weight aims only at attenuating the load worst case amplification.

C2- **Stability and roll-off of the controller**:

$$z_3 = T_3(K)y = W_k K y, \tag{51}$$

where

$$W_k(s) = \frac{s^2/\omega_k}{s^2/(g_k\omega_k)^2 + 2s/(g_k\omega_k) + 1}$$

is a high-pass filter, with positive parameters $g_k = 0.1$ and $\omega_k = 1000$ rad/s. These latter are also fixed.

Note that all parameters $\{g_e, \omega_e, t_r, g_k, \omega_k\}$ may be optimized but are chosen to be fixed in the process. Indeed, many other elements may be tuned.

Multi-channel optimization. Constraint C1 relates the tracking objective of a MLA function. C2 relates to the main load attenuation objective, while constraint C3 imposes controller stability and constraints its high-frequency responses (avoiding un-modelled and noise excitements in its output). This series of constraints T then reads

$$T = \textbf{blkdiag}(T_1, T_2, T_3) \tag{52}$$

and is the one to be optimized to find the appropriate K.

Controller structure. Concerning the set $\mathcal{K}$, the chosen controller structure is a dynamic output-feedback controller without direct feed-through term, i.e.

$$K : \left\{ \begin{array}{rcl} \dot{x}_c & = & A_c x_c + B_c y \\ u & = & C_c x_c \end{array} \right.$$

where A_c, B_c and C_c are matrices with appropriate dimensions defining a controller of rational order n_c to be determined and optimized in the MLA block. In addition, to ensure static gain tracking, an integral action is also imposed by adding the dynamic on $a_z^\star - a_z$.

"Optimization process". During the optimization, the only parameters to be optimized are n_c the dimension of the controller and γ the "optimality" of the performance. The former starts at a minimum value and the problem is solved with an objective $\gamma = 0$. Then, constraints are checked and iterations starts. The next section details this ad-hoc process as well as the analysis performed.

Analysis and iterations

Stability. After the initial optimization one obtains an attenuation level γ. If $\gamma > 1$, then the controller dimension is increased. Otherwise, the first property to be evaluated is the stability. This latter is easily checked on the ROM by analyzing the closed-loop eigenvalues. Then, in a second step, the stability is evaluated on the FOM including measurement and internal delays. This second step is performed either by approximating the delays via a Padé rational approximation or using a dedicated stability tool developed in [88], applicable to irrational functions. If the stability on the original model is satisfied, the performance is then analyzed.

Performance. Applying the controller on the full original model, and very that the weight constraints are satisfied, i.e. $\gamma < 1$. If true, then the controller has been found. Otherwise, increase the γ objective in the optimization, i.e. decrease the expected performance while keeping $\gamma < 1$.

The process is illustrated in the next part.

Illustration of whole MLA module

The proposed Matlab code reads as follows.

```
load('+flipased/ss_flexop_1_wing_gla_38')
load('+flipased/1wing_sym_gust_38_2021_3_25_18_45')
speed           = 38; % m/s
measDelay       = 200e-3;
trep            = 6; % MLA response time [seconds]
structure       = 3;
[K,CL,gam,info] = mla.main(sys,x_gust,speed,measDelay,trep,structure,false);
```

Note that the speed and measurement delays are the configuration parameters while `terp` is the t_r coefficient. `structure` is the original complexity of the model, n_c. Running the above code leads to:

```
------------------------------------------------------------------
>> CONSTRUCT MODEL
------------------------------------------------------------------
>> Select and merge input/output sets
>> Model informations
        * H2 unstable
        * 427 internal variables
        * 12 inputs, 7 outputs
        * 0 internal delays
        * 0 output delays
```

Which loads the model, set the input-output model without any delay. Then, on the rational form, the spurious poles are removed to avoid numerical issues. This performed as follows.

```
------------------------------------------------------------------
>> REMOVE SPURIOUS POLES
------------------------------------------------------------------
Should be (almost) zero: 1.06e-09
Should also be (almost) zero: 3.32e-08
Should be moderately large: 2.29e+01
Should be moderately large: 2.29e+01
>> Model informations
        * H2 stable
        * 426 internal variables
```

```
        * 12 inputs, 7 outputs
        * 0 internal delays
        * 0 output delays
```

Then, the internal and external delays are added to the model.

```
-------------------------------------------------------------------
>> ADD DELAYS
-------------------------------------------------------------------
>> Model informations
        * Internal delays, no stability check
        * 426 internal variables
        * 3 inputs, 4 outputs
        * 9 internal delays
        * 3 output delays
```

The resulting non rational model is then approximated to a rational form. Here, either the Robust Control Toolbox or the MOR Toolbox is used. The resulting model is finally normalized to the appropriately used in the control optimization step.

```
-------------------------------------------------------------------
>> RATIONAL APPROXIMATION AND ORDER REDUCTION
-------------------------------------------------------------------
>> Using Robust Control Toolbox (rational Pade approximation)
>> Model informations
        * H2 stable
        * 670 internal variables
        * 3 inputs, 4 outputs
        * 0 internal delays
        * 0 output delays
>> Model informations
        * H2 stable
        * 100 internal variables
        * 3 inputs, 4 outputs
        * 0 internal delays
        * 0 output delays
-------------------------------------------------------------------
>> CONTROL-ORIENTED MODEL (NORMALIZE)
-------------------------------------------------------------------
>> Unstable part size: 0
```

Now the model is ready for optimization. The loop starts and results on the single model investigated are reported here after.

```
-------------------------------------------------------------------
>> MLA LOOPS START
-------------------------------------------------------------------
        > OUTER-LOOP: optimize a control structure of order 3
        >> INNER-LOOP: optimize with objective gamma=0.00
                >> Compute weights
                >> Interconnect
                >> Construct MLA controller
Final: Peak gain = 1.19, Iterations = 244
```

```
                WARNING: no stability check possible
               -- Full closed-loop stable (score 100)
               -- Unsuccessful load attenuation control (gamma=1.19)
        > OUTER-LOOP: optimize a control structure of order 4
        >> INNER-LOOP: optimize with objective gamma=0.00
               >> Compute weights
               >> Interconnect
               >> Construct MLA controller
Final: Peak gain = 0.989, Iterations = 233
                WARNING: no stability check possible
               -- Full closed-loop stable (score 100)
               -- Unsuccessful load attenuation control (gamma=1.02)
        > OUTER-LOOP: optimize a control structure of order 5
        >> INNER-LOOP: optimize with objective gamma=0.00
               >> Compute weights
               >> Interconnect
               >> Construct MLA controller
Final: Peak gain = 0.859, Iterations = 503
                WARNING: no stability check possible
               -- Full closed-loop stable (score 100)
               -- Successful load attenuation control (gamma=0.90)
```

In this case, the controller dimension is increased from 3 to 5 and, at the end the controller of dimension 5 is able to provide stability and performance. The controller γ value is 0.859 on the reduced model and 0.9 on the full order one.

The MLA process presented in this section provides a simple way to compute such a function, with few parameters. It only requires the ASE model, the starting order of the controller. The rest is iteratively computed. The output of this computation is a controller for MLA $K(s)$ and a stability guarantee, together with an attenuation level γ. Of course, as mentioned in the introduction, every step may be amended or at least discussed. Still, when applied in the overall process, it allows generating the MLA function.

The MLA control results are shown in Figure 156.

Gust Load Alleviation (GLA) functions (DLR)

As part of the secondary control functions a gust load alleviation (GLA) controller is designed based on a single point reduced linear model. The overall goal is to reduce the maximum loads due to gust encounter by means of a controller. The starting point was the non-linear model developed within the flexop project. Besides the nominal inputs the model is extended by ten gust inputs. The aircraft is divided in ten gust zones along the aircraft longitudinal axis. Within each zone the aerodynamic panels of the experience the same gust velocity. This kind of modelling is an approximation, which reduces the complexitiy of the gust model strongly, while the effect of the gust on the aircraft is almost unaffected. In order to analyse the performance of the gust load alleviation controller load outputs at the wing roots and at the V-tail roots are provided.

As a first step the GLA controller should be designed at a velocity of 38 m/s and an altitude of 800 m, for which the non-linear model is linearized. The gust is considered a vertical 1-cos gust, like shown in Figure 157, that hits the aircraft symmetrically starting at the nose.

The gust velocity is given by U_0 and the gust half length by H. With increasing time the gust zone moves to the aft of the aircraft. In each gust zone the corresponding aerodynamic panels are affected by the gust speed, that is observed at the front edge of the gust zone. Namely, within a gust zone the gust speed is constant. The difference of gust speeds in two neighbouring gust zones is defined by a time

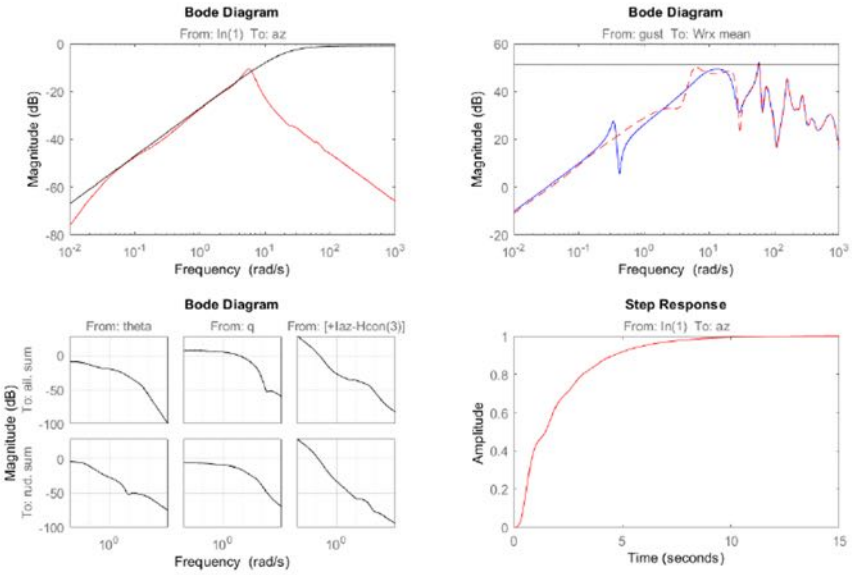


Figure 156: MLA control results

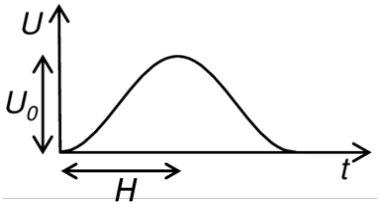


Figure 157: Gust illustration

delay. As a transfer function a delay can be defined by

$$G_{delay}(s) = e^{-t_{delay}s} \tag{53}$$

where t_{delay} is the time delay in seconds and s is the Laplace variable. To simplify the handling of time delay, it is approximated by a second order Padé approximation

$$G_{delay}(s) \approx \frac{s^2 - \frac{6}{t_{delay}} + \frac{12}{t_{delay}^2}}{s^2 + \frac{6}{t_{delay}} + \frac{12}{t_{delay}^2}} \tag{54}$$

The selected control synthesis method is the structured H-infinity one. It solves the optimisation problem

$$\min || T_{w \to z}(K) ||_\infty, \tag{55}$$

$$K \in \mathcal{K} \tag{56}$$

for which the H_∞ norm of the closed transfer function T is minimized, while the structure of the controller is predefined. Here, the controller is considered to be a simple gain matrix. Like shown in Figure 158 the inputs to the controller are the pitch angle y_θ, the pitch rate y_q, the z-acceleration measured in the fuselage $y_{az_{fu}}$ and the z-accelerations at the wing tips $y_{az_{wl}}$ and $y_{az_{wr}}$.

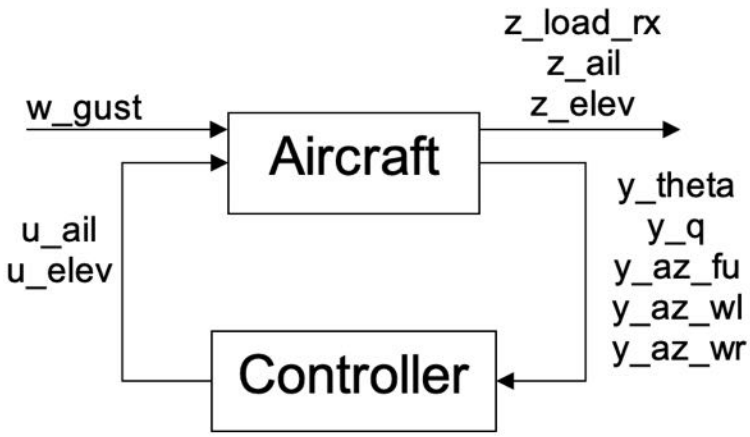


Figure 158: Closed-loop scheme

Based on these measurements the controller provides the deflection of the outer ailerons u_{ail} and all elevators u_{elev}. As the aircraft is almost symmetric along the longitudinal axis and only vertical gust encounters are considered, the deflection of the two ailerons are identical as well as the deflection of the four elevators. Therefore, the commands of the controller can be combined to two signals. The inputs and outputs of the aircraft state-space system used for the GLA control synthesis are also normed for a better numerical handling. At this point the state-space system has more than 400 states. To reduce the order of the system in a numerical way the balanced reduction is used to decrease the system order to 60.

Before the structured H-infinity synthesis can take place, the requirements of the control problem have to be defined. Three different requirements are defined for the GLA controller synthesis. Firstly, the H_∞ norm of the weighted transfer function from gust to the wing root bending moment should be minimised. The transfer function is multiplied with a weighting function to emphasize for which frequency domain the wing root bending should be reduced especially. Secondly, the action of the aileron and the elevator actuators is limited in deflection and deflection rate for GLA controller. Additionally no interaction of the GLA controller with the flight dynamics is wanted as well. This leads to requirements with respect to the transfer function from gust to the aileron and elevator deflections. As the maximum deflection rate of the ailerons and elevators differ, two requirements are defined in Figures 159-161 show the defined requirements (black), the open-loop (magenta) and closed-loop (blue-red) transfer functions.

The deflection of the ailerons and the elevators stays within the predefined bounds, while the wing root bending can be reduced with a GLA controller in a frequency range of approximately 2 - 11 rad/s. Furthermore, the maximum peak of the open-loop system at 58.4 rad/s is reduced significantly. At various frequencies the closed-loop wing root bending moment might exceed the one of the open-loop case, but anyways the maximum value is reduced.

Time simulations of the different gust excitations show the reduction in the maximum peak load as well. Figure 162 shows the wing root bending to a step excitation at the gust input.

It is visible, that the maximum load is reduced by almost 10%. In addition, two 1-cos gusts refering to a frequency of 58.4 and 25.6 rad/s are considered. For a frequency of 58.4 rad/s, at which the maximum wing root bending for the open-loop case is reached, the simulation shows only a 3% reduction of the

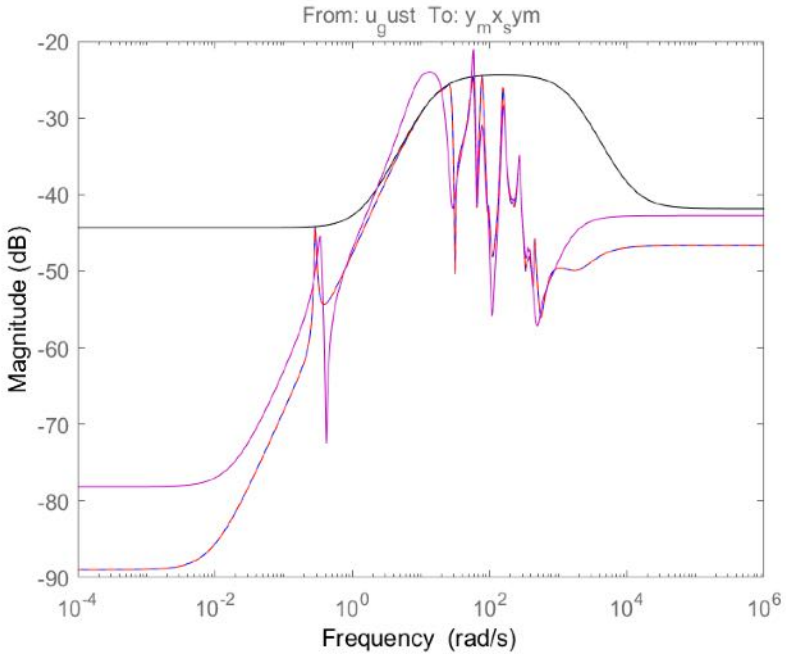


Figure 159: Bode plot of the transfer function from w_{gust} to $z_{load_{rx}}$

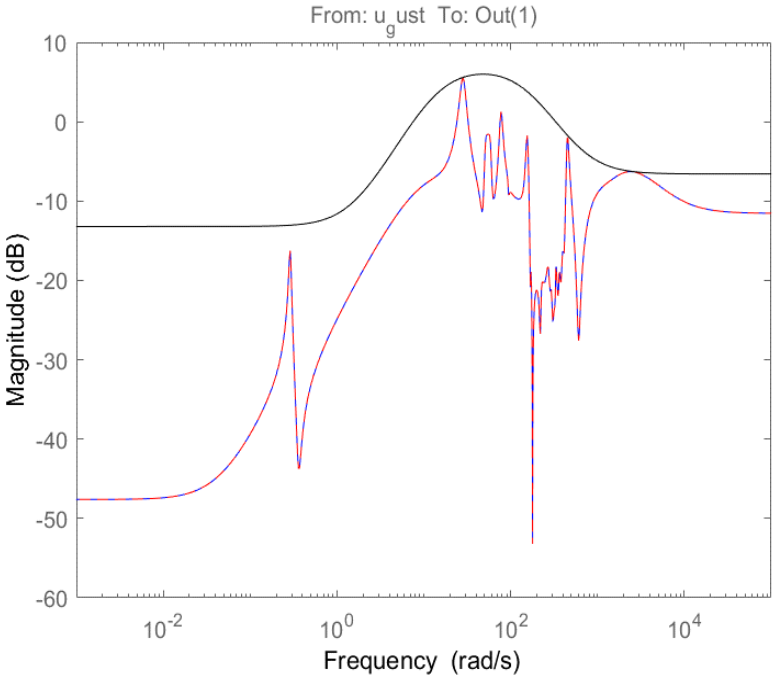


Figure 160: Bode plot of the transfer function from w_{gust} to z_{ail}

maximum load in Figure 163.

As the excitation by a 1-cos gust cannot be restricted to a single frequency, it is possible, that the load reduction is not as high as expected. For a critical gust half length defined by Pratt's method

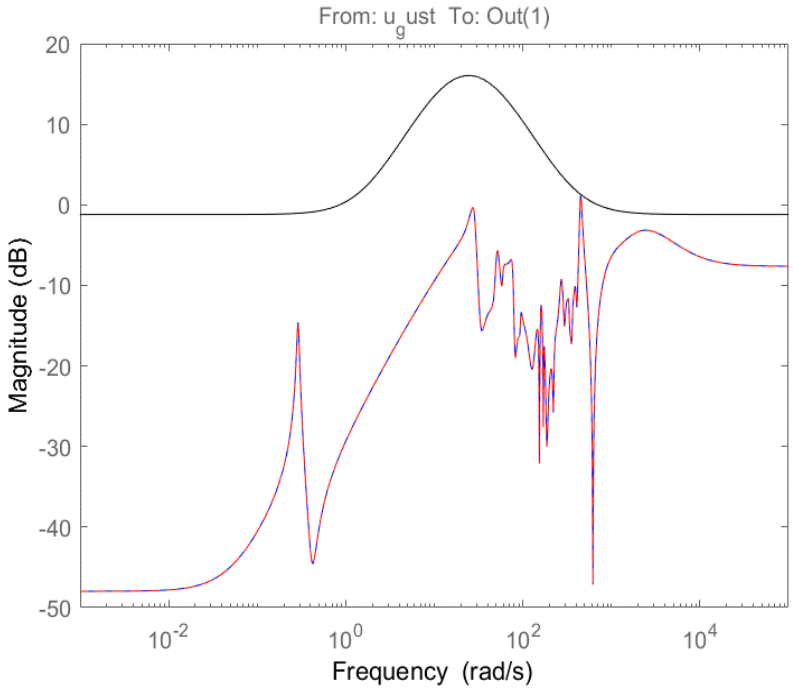


Figure 161: Bode plot of the transfer function from w_{gust} to z_{elev}

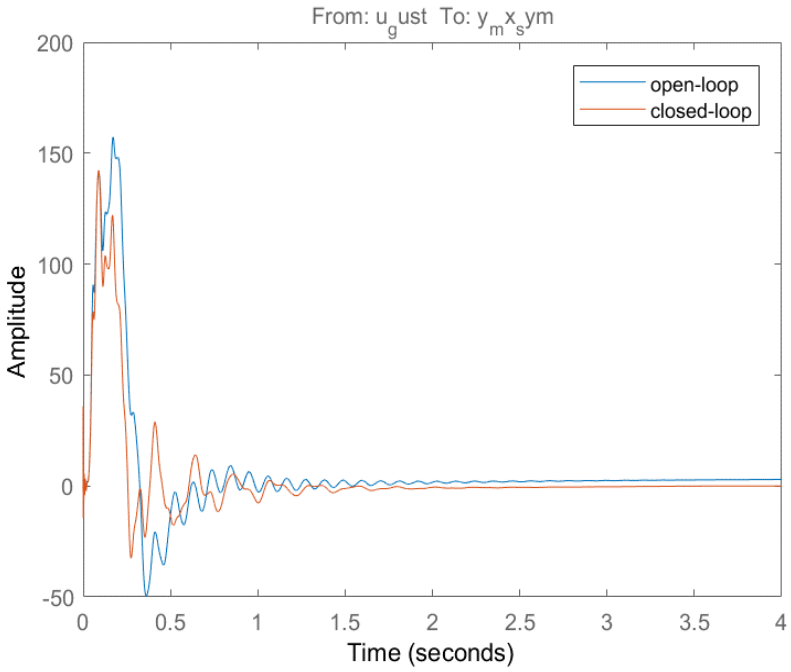


Figure 162: Step response from w_{gust} to $z_{load_{rx}}$ [Nm]

$$H_{crit} = \frac{25}{2} c_{ref} \tag{57}$$

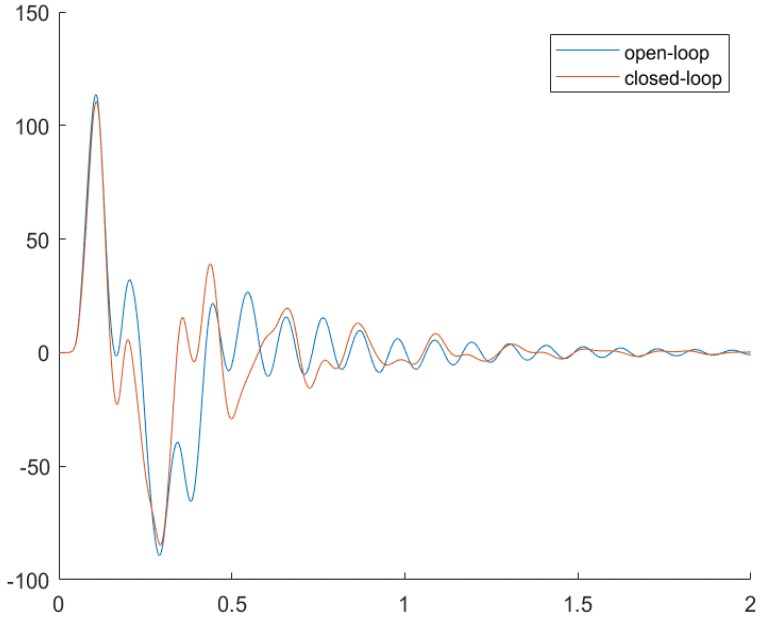


Figure 163: $z_{load_{rx}}$ [Nm] due to 1-cos gust (58.4 rad/s)

where c_{ref} is the reference chord length, a nominal 1-cos gust excitation of 25.6 rad/s is achieved. The time simulation for this excitation is shown in Figure 164.

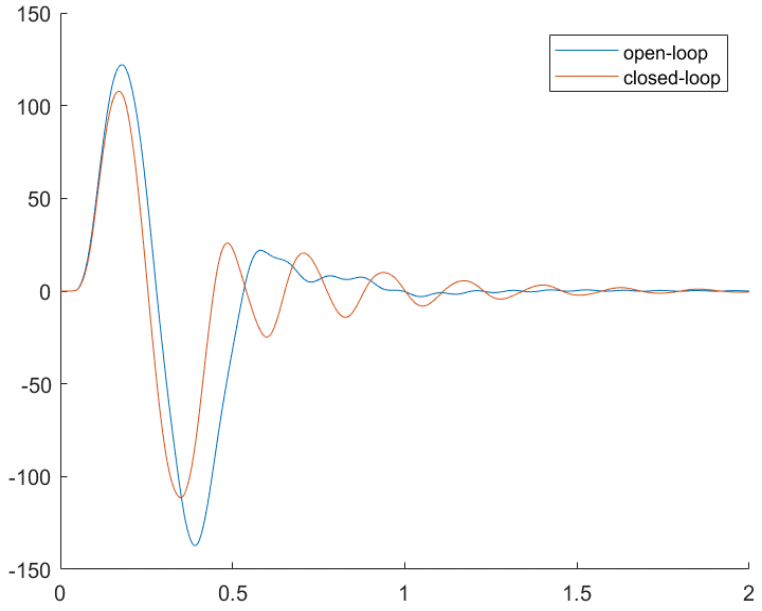


Figure 164: $z_{load_{rx}}$ [Nm] due to 1-cos gust (25.6 rad/s)

Here the maximum wing root bending is reduced by almost 12%. However, the wing tends to vibrate longer. For now, the synthesised GLA controller is considered sufficient for an integration in the overall

toolchain of the project. Therefore, the next step will be the implementation of the GLA control synthesis algorithm into RCE.

Flutter control design functions (SZTAKI)

The flutter controller design is done based on the uncertain LPV ASE model of the aircraft. Since rigid body modes might be unstable and the aim of the flutter controller is to stabilize the flutter modes, the inner loops of the baseline controller are connected to the model. The flutter control design takes the outer aileron (denoted by L4 and R4) actuator dynamics and the flutter control design model FlexAC-Model as inputs via CPACS.

The internal structure of the flutter controller is shown in Figure 165. The flexible model of the aircraft is split into the lateral and longitudinal model. The two flutter modes appear separated in these two system, hence the this separation allows us to stabilize them one-at-a-time. The controller designed to stabilize the symmetric (longitudinal) and asymmetric (lateral) flutter mode is denoted by K_{sym} and K_{asym} respectively. Both of these controllers are SISO and they are both augmented by a low pass filter

$$F(s) = \frac{1.6 \cdot 10^5}{s^2 + 560s + 1.6 \cdot 10^5} \tag{58}$$

to limit their bandwidth. The input of the flutter controller consists of the pitch rate (q), and angular rate measurement from the L6 and R6 IMU sensors (q_{L} and q_{R}). The actuating signals are the deflection commands for the pair of outermost ailerons in Figure 145 ($u_{\text{f,L}}$ and $u_{\text{f,R}}$). These signals are blended together as depicted in Figure 165.

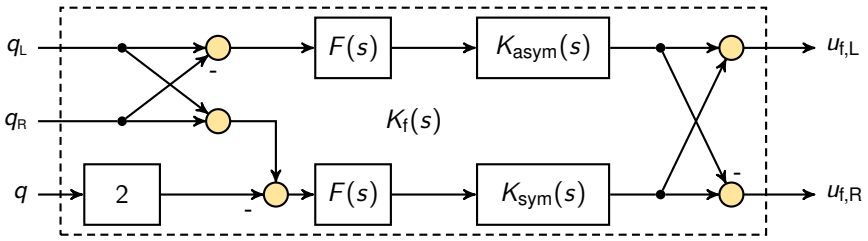


Figure 165: Internal structure of the flutter controller.

The design of both K_{sym} and K_{asym} is carried out for the nominal value of the uncertainties in the structural dynamics. Two values of the airspeed are selected: 38m/s and 60m/s. The controllers are designed for these two values simultaneously with structured H_∞ synthesis. The generalized plant interconnection is illustrated in Figure 166. Here, the weighting functions are $W_e(s) = 1/2$, $W_u(s) = 1/10^\circ$, and $\tau(s)$ is the forth order Padé approximation of 15ms delay. The objective of the synthesis is to minimize the sensitivity function of the closed-loop which results in robust stabilization. For a detailed reasoning for and explanation of this design process see [85]. The MIMO flutter controller is then formed by the interconnection of SISO controllers and the filters as in Figure 165.

Similarly to the baseline control design algorithm, the flutter suppression control design block needs to be augmented with basic analysis algorithms to verify if the resulting controller satisfies the control performance specifications. As a main measure, the multi-input multi-output (MIMO) disc margins are selected. 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.

Analysis of the designed (flutter) controllers, prior RCE integration (SZTAKI, DLR)

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

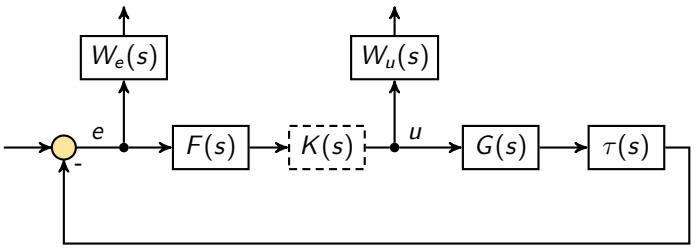


Figure 166: Generalized plant interconnection used for the flutter control design.

at which the closed-loop becomes unstable is computed in each channel, simultaneously. First, the robustness of the baseline controller is analyzed without the flutter controller. The speed at which the disk margins become zero is considered the open-loop flutter speed. In the next step, the flutter controller is also connected to the system and the margins are recaclulated. This step reveals how much the flutter controller is able to extend the safe flight envelope functioning simultaneously with the baseline controller.

This procedure shall be extended to GLA and MLA.

Furthermore, different methods of flutter analysis are applied for the open- and closed-loop model, in order to judge the benefit of the flutter control and validate the used tools. A Simulink model representing the nonlinear flexible aircraft dynamics can be used in order to determine the open-loop flutter speeds and frequencies. The aircraft is trimmed and linearised for a couple of flight conditions. Especially, differences in flight speed are of interest for the demonstrator aircraft, as they have the biggest effect on the aeroelastic modes. The set of linearised state-space models is then analysed with respect to their eigenvalues. It is then possible to see the gradual change in the eigenvalues and therefore the frequency and damping for varying flight conditions as exemplary shown in Figure 431.

As soon as a pole crosses the imaginary axis and migrates to the right half plane, unstable flutter becomes an issue. Based on the trimmed airspeed of the linearised systems it can then be determined what the flutter speed is. This leads then to Figure 168, which shows the damping and flutter frequency for symmetric and asymmetric flutter over the flight speed.

As soon as the damping crosses the zero line the corresponding aeroelastic mode becomes unstable.

Closed loop analysis block (SZTAKI)

The final block of the MDO toolchain is the analysis block. In the present case, the performance of the baseline and flutter controllers are evaluated. For the closed loop analysis the baseline and flutter controllers are connected with the flexible aircraft model. Figure 155 shows the pole migration of the FlexACModel with (red) and without (blue) the baseline controller. The designed baseline controller is stable with the flexible model up to the flutter speeds.

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. First, the robustness of the baseline controller is analyzed without the flutter controller. The speed at which the disk margins become zero is considered the open-loop flutter speed. In the next step, the flutter controller is also connected to the system and the margins are recaclulated. This step reveals how much the flutter controller is able to extend the safe flight envelope functioning simultaneously with the baseline controller. Figure 169 shows the disk margins obtained by this analysis. For a more detailed explanation the reader is again referred to [85].

The output of the analysis block are the open loop, the absolute and robust flutter speeds.

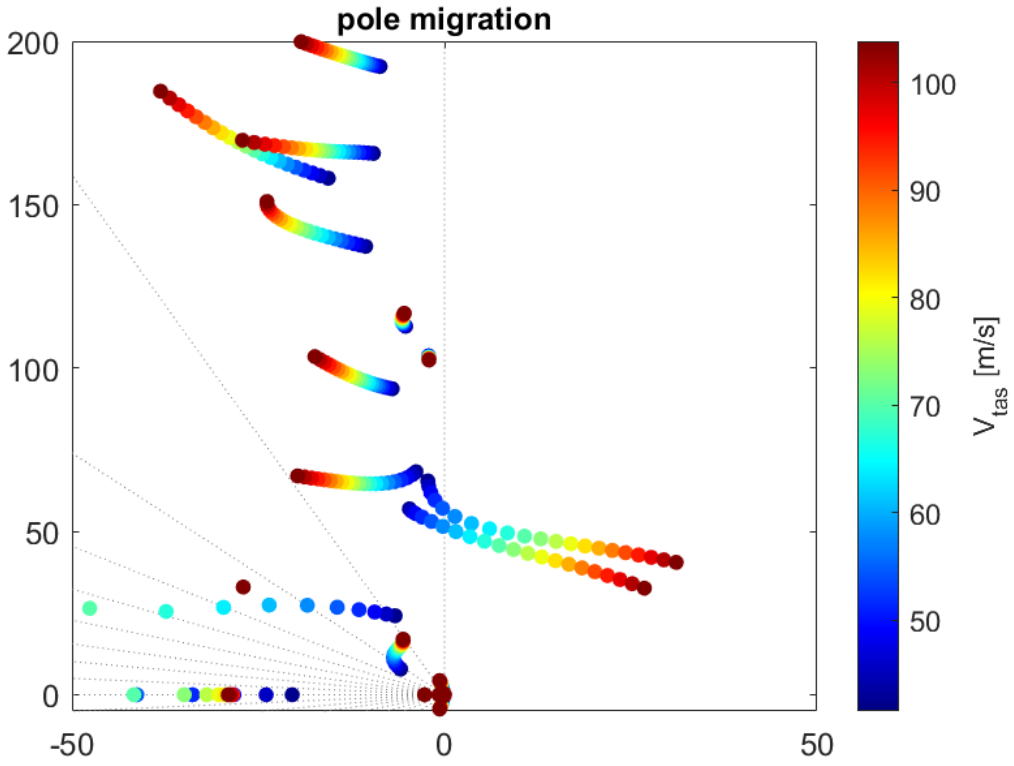


Figure 167: Pole migration with respect to the true airspeed

Integration of the baseline controller design in the MDO workflow (SZTAKI)

The design process of the baseline controller is carried out on the basis of the mathematical description for the aircraft. Structurally the controller consists of several loops targeting different dynamical modes. Accordingly, intuitive design specification for the loops can be formulated by the user in terms of settling times, reference tracking or robustness margins. The control design itself automatically optimizes the corresponding gains, in order to satisfy the specified design goals. Once the optimization found a feasible solution it provides the corresponding control gains and control structure which is then used for the numerical analysis. However, a simple metric is also returned for the user which indicates the performance of the control loops. This allows the interaction with the automated design process: the user can formulate tighter or loser specifications according to the individual needs. A clear graphical representation is also provided which can be included in the reporting. In addition, the controller generation process adjust the speed-dependence of the control gains in order to achieve the best possible performance and the simplest scheduling function. Frequency and time domain results can be seen in Figure 170.

RCE with Modeling and Control design blocks (SZTAKI)

The performance evaluation is done in two steps. First, it is critical to evaluate the RCE implementation. This is presented for the modeling and control design blocks for the baseline and flutter suppression control design. The RCE implementation of these two blocks is shown in Figure 171.

First, an 'Input Provider' is used to send the initial CPACS file, then the Modeling component start processing and sets an output based on the actual modelling script. The output is forwarded to the

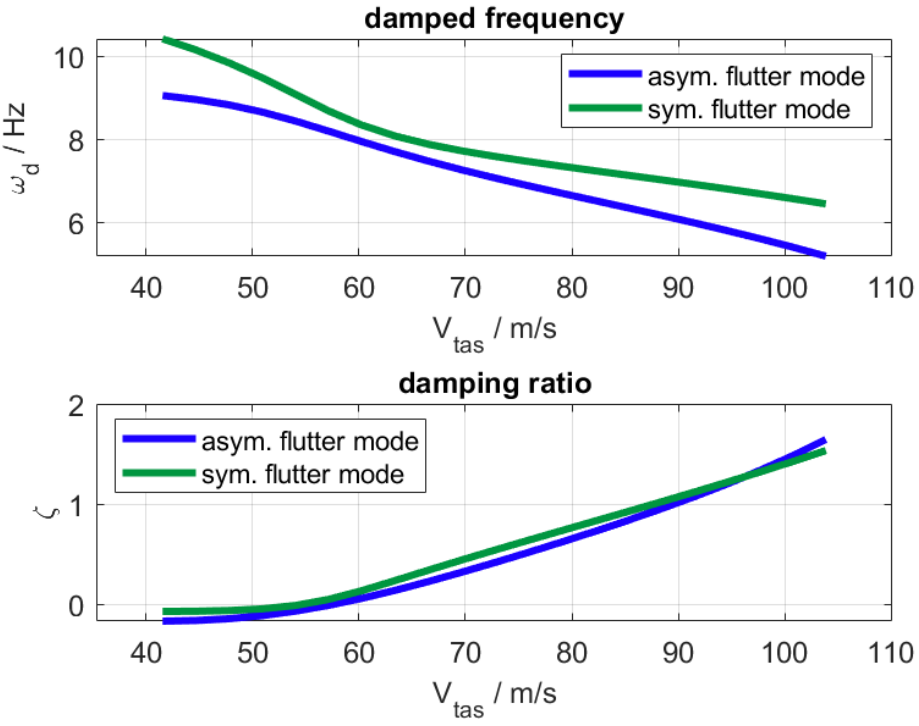


Figure 168: Frequency and damping with respect to the true airspeed

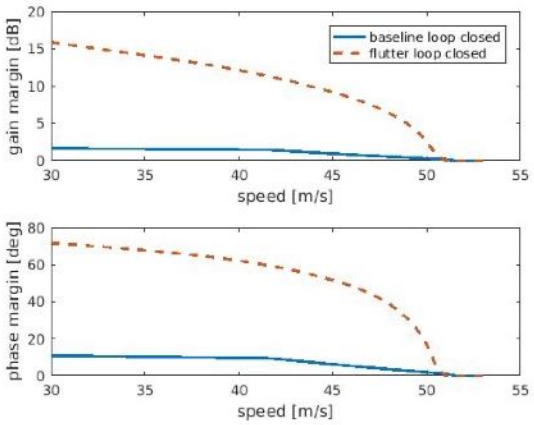


Figure 169: Resulting disk margins

Baseline (RigACModel) and Flutter (FlexACmodel) Controller components. Once the baseline design is finisehd, the baseline design gains are sent to the flutter design block. Finally, the Analysis block receives the flexible model, the baseline gains and the flutter controller. The modeling, he control design components and the analysis tool function with a help from external scripts which act like wrappers between them and the actual Matlab files. The scheduling between the blocks is based on the data that is the output of the preceding block. The output is set using the post-execution commands of the modeling block. The output is written in the output directory in accordance with the wrapper, so when

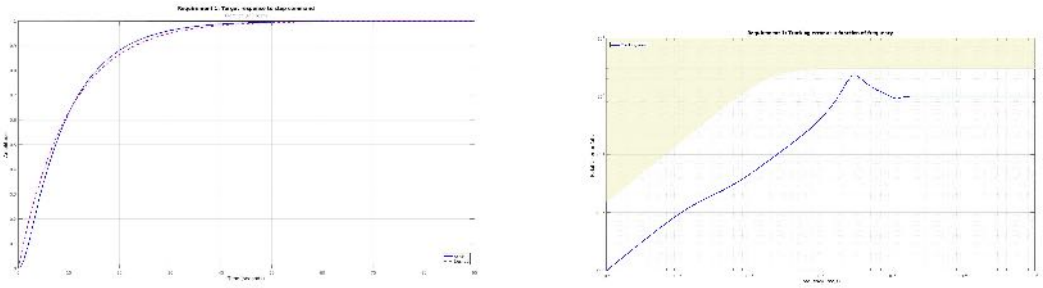


Figure 170: Time domain results (left) Frequency domain results (right)

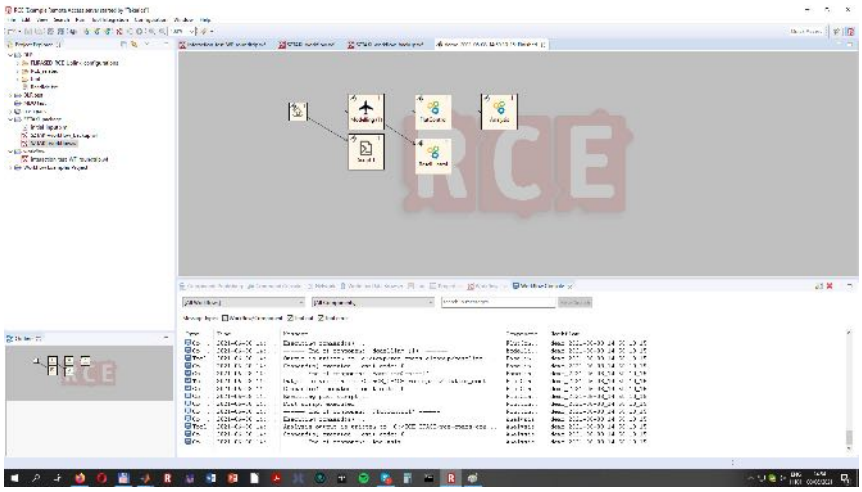


Figure 171: RCE implementation of the modeling and flutter control design blocks

the current block finishes, the post execution commands are executed.

All RCE block communications and data sharing needs to be specified in addition to the scheduling of the RCE blocks. The control oriented modeling blocks output files are referenced in CPACS. These output files are given in the ToolSpecific field of the CPACS xml file.

The second step is to evaluate the results of the control design blocks. This step is carried out for each controller individually first. For the baseline controller the first step is to evaluate if the handling qualities are satisfied or not. If this can not be achieved by the resulting controllers then the handling qualities need to be relaxed. In addition to the handling qualities, robustness, gain and phase margins of the resulting controller is evaluated. The analysis results are also written in the corresponding ToolSpecific field of the CPACS xml file. The flutter controller is also analyzed if it satisfies the robustness analysis criteria.

Validation of the integrated design toolchain for collaborative design

The objective of the validation is to conduct a comprehensive comparison of results and findings obtained from various sources. The primary purpose of this specific assessment is to instill confidence in the developed tools and methods for the collaborative design toolchain.

The data acquired from flight tests will serve as a reference point for validating models related to structural dynamics, aerodynamics and controls. To facilitate this validation process, analysis tools will be

designed for test and simulation results. For the structural dynamics tools will be available in Nastran which can be tuned based on available ground testing data. For the aerodynamics the output of different tools will be compared with flight test data gathered for the demonstrator aircraft. Furthermore, the results of the different methods available for flutter analysis will be cross-compared in order to gain confidence in the aeroelastic modeling at different stages of the collaborative design toolchain.

The proposed approach of model validation allows a thoroughly examination of the tools and methods, which are chosen for the collaborative design toolchain.

Connection between MDO toolchain and testing

The HIL test and flight test blocks play crucial roles in validating the developed methodologies, as depicted in Figure 172.

The HIL tests focus on evaluating the practical implementation aspects of the controllers, serving as a final step before conducting flight tests.

In the case of the MDO toolchain, flight tests serve two primary objectives. Firstly, they validate the maturity of the control design technology, particularly for the manoeuvre load alleviation (MLA), gust load alleviation (GLA), and flutter suppression controllers, which have not undergone flight testing using the model-based design methodology within FLiPASED. Secondly, flight tests provide an opportunity for fine-tuning the controllers manually to achieve optimal performance and gain valuable insights for both the designers (making the models and the controllers) and the broader aviation community. In this context, the automatic execution of the synthesis algorithms is not a critical criterion, as the focus lies on optimizing controller performance based on the updated aircraft model derived from flight test data.

At the conclusion of the cycle, the lessons learned from the HIL tests and flight tests are fed back into the MDO toolchain through engineering considerations. If the HIL tests identify any implementation challenges with specific controllers, the corresponding control design algorithms need to be updated. Similarly, if the flight tests reveal performance or robustness issues with a controller, the algorithms must be adjusted accordingly.

Structural dynamics model validation (DLR-AE)

The tasks related to structural dynamics of the aircraft models are led by DLR-AE, but contributions are made by ONERA, TUM, SZTAKI and DLR-SR as well.

The main steps regarding the task are:

- structural model development and ground vibration test (GVT) based update
- Model comparison and fine tuning for RCE toolchain based and GVT based model matching
- Operational modal analysis based model update during flight tests and its connection how this feeds back to NASTRAN models
- Description of used tools and how they can we standardized

In this chapter, a summary of the structural dynamics model and the model-updating activities pertaining to its update are described.

Comparison of -1 aircraft structural dynamic model with static test

The FLEXOP project conducted a comprehensive static test of the -1 wing simultaneously with the -0 and -2 wings. The primary objective was to verify the stiffness properties of the manufactured wing and validate the accuracy of the FE model developed during the design stage.

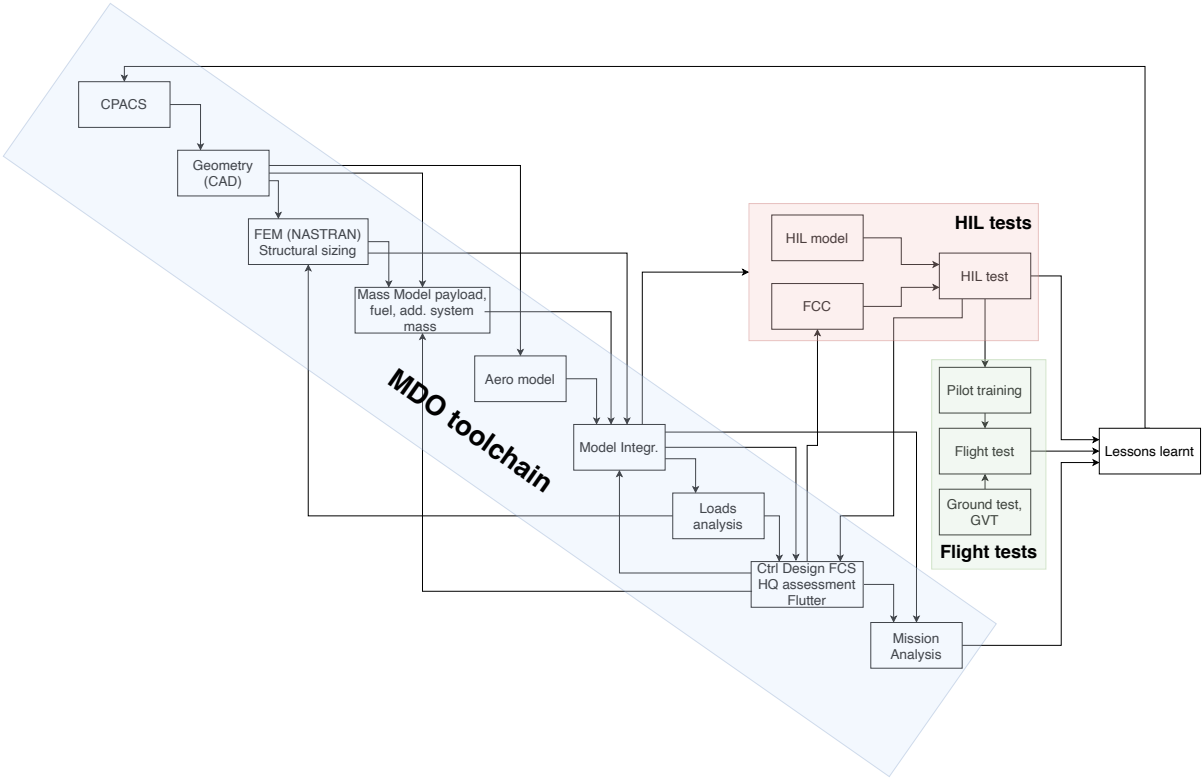


Figure 172: Toolchains developed in FLiPASED

Figure 173 illustrates the wing tip deflection at different load cases, showcasing the linear relationship between the applied load and deflection. However, it is important to note that due to measurement errors, a zero drift was observed when the load was increased from zero and then decreased back to zero.

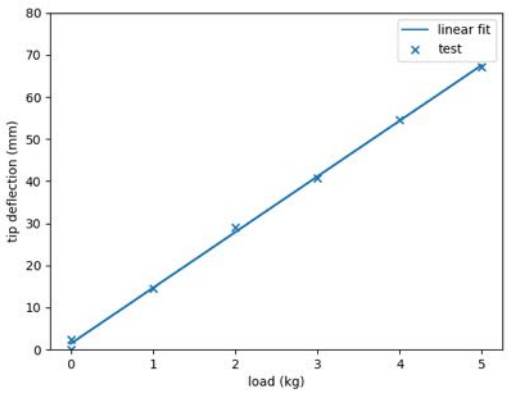


Figure 173: Displacement vs load at tip of the wing from static tests (-1 wing)

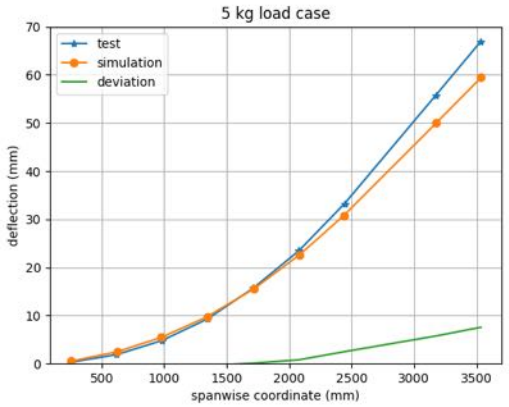


Figure 174: Span-wise displacement of wing under tip load (-1 wing)

To replicate the static test, the FE model was refined and adjusted. A comparison between the simulation and the actual test was conducted, focusing on the span-wise displacement of the wing under a

5 kg tip load (Figure 174). The results revealed that the manufactured wing exhibited greater flexibility than initially modeled, showing a similar trend to the -0 and -2 wings. The deviation between the simulation and test results was approximately 12%, not accounting for the zero drift observed during the test.

Furthermore, the torsional load cases were also investigated. Figure 175 demonstrates the linearity of the model under various torque loads. A comparison of the span-wise torsion of the wing under a 2 kg torque load (Figure 176) revealed minimal differences of only 0.1 degrees. Taking into account the inherent measurement errors, the simulation and test results aligned quite well.

These static test findings provide valuable insights into the stiffness properties and structural behavior of the -1 wing. The significant flexibility observed in the manufactured wing highlights the importance of real-world testing and serves as a basis for further design refinements. By enhancing the accuracy of the FE model and addressing the observed deviations, future optimizations can be made to improve the wing's performance and ensure its reliability during flight operations.

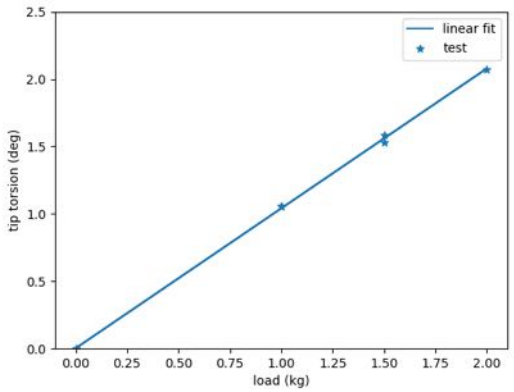


Figure 175: Torsion vs load at tip of the wing from static tests (-1 wing)

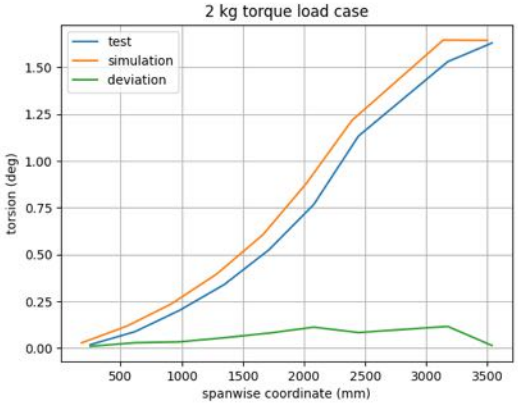


Figure 176: Span-wise torsion of wing under tip load (-1 wing)

Comparison of -1 aircraft structural dynamic model with GVTs

The generation of the -1 wing FE model was carried out at TUM using a CATIA-Hypermesh toolset. This highly detailed model encompasses both structural and non-structural components, including on-board systems, ensuring a high-fidelity representation.

Table 18 presents a comparison of the eigen frequencies between the -1 aircraft model (without updates) and the ground vibration test (GVT) results. Generally, a good agreement is observed between the FE model and the GVT outcomes. However, two significant observations arise from this comparison.

Firstly, the third flexible mode (3n_wing_bending-a) exhibits the most notable difference between the experimental results and the GVT. Given that this wing bending mode plays a critical role in the flutter mechanisms of the -1 aircraft, it becomes crucial to update the wing FE model to accurately capture the frequency of this mode.

Secondly, the second flexible mode (1n_wing_in-plane-a) is observed during the GVT but not in the FE simulations. This mode involves relative motion between the fuselage and wing, as depicted in Figure 177. This occurrence can be attributed to some degree of free-play or softness in the attachment between the fuselage and wings, which is not accounted for in the idealized attachment assumed by the FE models. To simulate this mode, an approach under consideration involves introducing soft

Mode	GVT (hz)	FE (hz)	Δf (%)
2n_wing_bending-s	2.94	2.91	-1.02
1n_wing_in-plane-a	7.01	–	–
3n_wing_bending-a	7.57	8.15	7.66
wing_torsion-s	10.27	10.50	2.24
wing_torsion-a	10.73	10.61	-1.12
4n_wing_bending-s	12.13	12.11	-0.16
2n_wing_in-plane-s	15.07	15.06	-0.07

Table 18: Comparison of eigen frequencies of the flexible modes: GVT vs FE model of the -1 aircraft (*i*n - *i* nodes in the mode, s - symmetric, a - antisymmetric)

springs at the wing-fuselage interface to ensure its presence in the simulation. Furthermore, ongoing studies are exploring the update of the FE model to address the aforementioned mode, and the use of tuning beams is planned as part of this endeavor.

In conclusion, the comparison between the -1 wing FE model and the GVT results has highlighted the need for updates to accurately capture the critical wing bending mode and address the relative motion between the fuselage and wing. These updates, along with the incorporation of tuning beams, will contribute to improving the fidelity of the FE model and enhance its ability to simulate the aircraft's behavior more accurately.

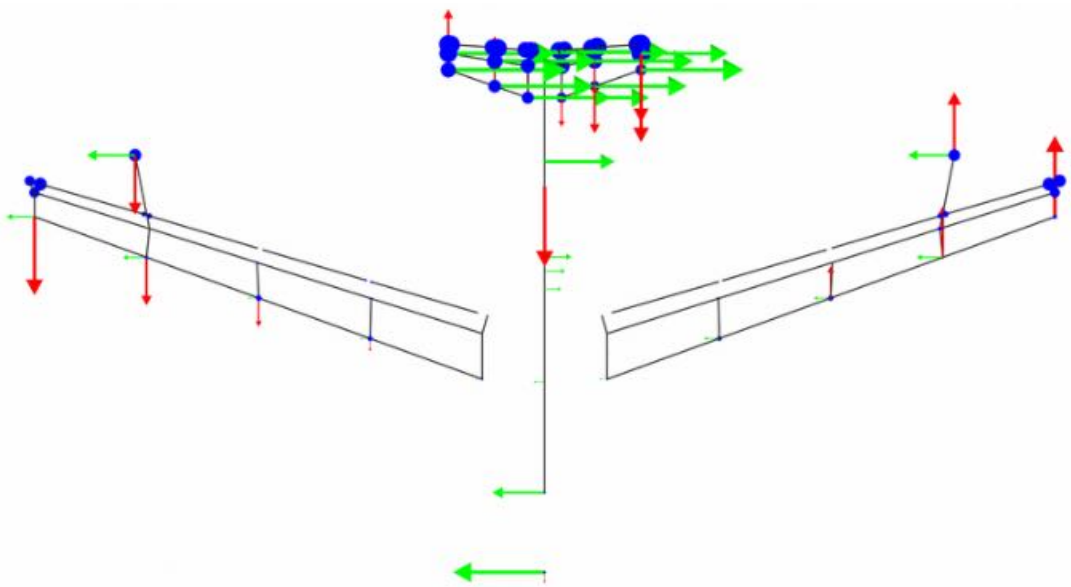


Figure 177: -1 aircraft 1n_wing_in-plane mode

Model-updating of the -1 wing

The initial model updating of the -1 wing is performed using data obtained from the static test. In this process, a knock-down factor is applied to the engineering stiffness values (E_1, E_2, G_{12}) of the wing skin and spar. The model updating procedure focuses on the 3 kg bending load case as a basis. As depicted in Figure 178, the simulation results exhibit a close resemblance to the test data. The deviation between the simulation and test results is reduced to 2mm, falling within the range of test error.

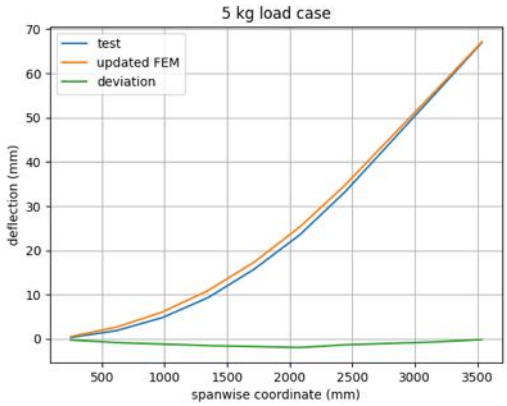


Figure 178: Span-wise displacement of wing under tip load (-1 wing)

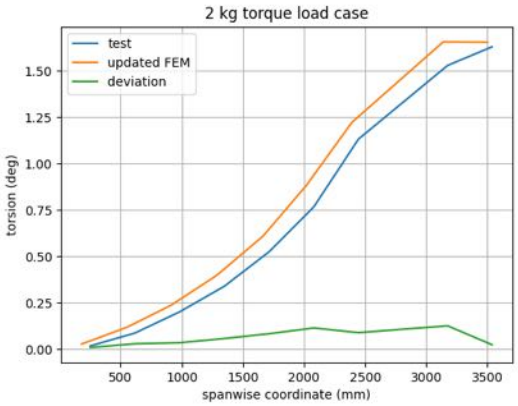


Figure 179: Span-wise torsion of wing under tip load (-1 wing)

Moving forward, the torsional load case with a 2kg load is simulated using the updated model, as illustrated in Figure 179. As expected, no noticeable differences are observed since the parameter updating primarily accounts for the bending load case. With the completion of the model updating process, a modal analysis is conducted using the updated model. Figure 180 presents a comparison of the eigenfrequencies among the ground vibration test (GVT) data, the original FE model, and the updated FE model.

Notably, only the 3n asymmetric wing bending mode exhibits improvement, while all other modes show a deterioration. This outcome can be attributed to the fact that the static test-based updating primarily tunes down the engineering stiffness (E_1), consequently resulting in decreased eigenfrequencies. To address this issue, the next step involves the localized implementation of tuning beams to enhance the performance of the 3n bending mode without adversely affecting the other mode shapes.

Comparison of RCE aircraft model with static test and GVT

The initial model generated through the MDO toolchain was developed to simulate the static test setup. However, it was observed from Figure 181 that the RCE model exhibited significantly higher stiffness compared to the manufactured wing. To address this discrepancy, a similar approach as the -1 wing updating process was employed, applying a knock-down factor to the engineering stiffness of the wing spar and skin. As illustrated in Figure 182, the results of the wing bending simulation with the updated RCE model show a much closer alignment with the static test results.

Additionally, a modal analysis was conducted for both the initial RCE model and the updated RCE model. The outcomes of this analysis can be observed in Figure 183. Notably, following the update, improvements were observed in all the listed modes, indicating a better alignment between the simulation and test data.

Results of the MDO toolchain parameter study (TUM, DLR, SZTAKI)

Nr	Mode	GVT	FEM	deviation %	updated FEM	deviation updated %
0	2n wing bending-s	2.94	2.91	-1.19	2.74	-6.91
1	3n wing bending-a	7.57	8.15	7.7	7.66	1.13
2	wing torsion-s	10.27	10.5	2.2	10.43	1.6
3	wing torsion-a	10.73	10.61	-1.15	10.53	-1.86
4	4n wing bending-s	12.13	12.11	-0.19	11.42	-5.83
5	2n wing inplane-s	15.07	15.06	-0.05	14.32	-4.96

Figure 180: Comparison of eigen frequencies of the flexible modes: GVT vs FE model vs updated FE model of the -1 aircraft (*i*n - *i* nodes in the mode, s - symmetric, a - antisymmetric)

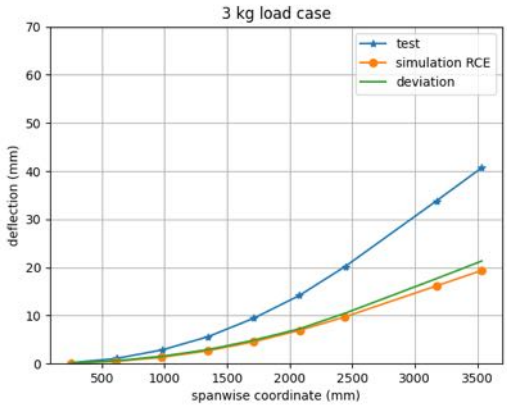


Figure 181: Span-wise bending of wing under tip load (-1 wing RCE model)

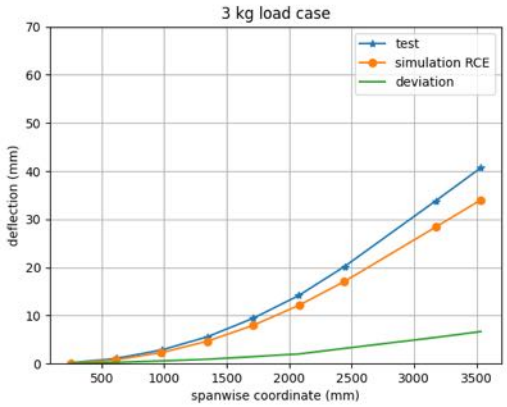


Figure 182: Span-wise bending of wing under tip load (-1 wing RCE updated model)

Nr	Mode	GVT	RCE FEM	deviation %	updated RCE FEM	deviation updated %
0	2n wing bending-s	2.94	3.81	29.68	2.89	-1.54
1	3n wing bending-a	7.57	10.05	32.82	7.8	2.99
2	wing torsion-s	10.27	12.25	19.3	10.65	3.74
3	wing torsion-a	10.73	12.7	18.33	10.86	1.24
4	4n wing bending-s	12.13	17.38	43.31	13.25	9.25
5	2n wing inplane-s	15.07	17.86	18.54	14.43	-4.27

Figure 183: Comparison of eigen frequencies of the flexible modes: GVT vs RCE FE model vs updated RCE FE model of the -1 aircraft (*i*n - *i* nodes in the mode, s - symmetric, a - antisymmetric)

The parameter study is set up in the following way:

- Sweep angle: 0 to 30 degrees
- Flutter mass: 0 to 0.4 kg
- Ply angle: -45 to 45 degrees

The MDO setup is set up based on the configuration presented in this document. All parameter cases run through in RCE with one study taking around one hour computational time. The first lessons from the parameter study are in correlation with the expectations:

- Higher flutter mass drives flutter speed down, 16-18% flutter speed increase possible with active control
- Sweep angle: low values do not have significant values, larger than 25 degrees makes active control difficult

Note: the models were created up to 70 m/s airspeed.

- 0 kg flutter mass: resulted in stable flutter modes up to 70 m/s airspeed
- .12 kg flutter mass: the closed loop is robustly stable up to 70 m/s airspeed: the closed loop could be stable for higher speeds as well

The effect of the sweep angle and flutter mass on the open loop, robust closed loop and absolute closed loop flutter speeds is given in Table 19.

Table 19: RCE results

Mass [kg]	Sweep [deg]	Open loop flutter [m/s]	Robust flutter [m/s]	Increase [%]
0.24	5	53	62	1.1698
0.24	12.5	59	67	1.1356
0.24	17.5	51	60	1.1765
0.24	22.5	63	67	1.0635
0.24	27.5	55	64	1.1636
0.06	20	70	70	1
0.18	20	60	64	1.0667
0.3	20	52	62	1.1923
0.42	20	46	56	1.2174
0.36	20	49	59	1.2041
0.24	20	55	65	1.1818

Wing shape control for minimization of induced drag (TUM, DLR, SZTAKI)

The present section focuses on drag optimization based on optimal aileron deflections. A summary of some of the tools used in the present induced drag modelling and optimization exercise is presented in Table 20.

Trefftz plane implementation
A far-field Trefftz plane implementation similar to the one presented in [58] is implemented. The induced velocity and drag are calculated on a plane downstream of the aircraft known as the Trefftz plane, using the span-wise lift distribution projected onto the Trefftz plane as shown in Figure 184.

Table 20: Summary of tools considered in the drag modelling and optimization exercise
(In the Feature column, 1. refers to the form of aeroelastic formulation, 2. refers to the method of estimating induced drag)

Tool	Aerodynamic method	Optimizer	Feature
NASTRAN aeroelastic solver	DLM	• Kriging-Regression model • SciPy *optimize*	1. Closely-coupled 2. Trefftz plane
VarLoads	VLM	• MATLAB *fmincon*	1. Closely-coupled 2. Surface-pressure integration
PANUKL	VLM, 3D panel	• Surrogate model • MATLAB *fminsearch*	1. Loosely-coupled (iterative) 2. Trefftz plane
AVL	VLM	• NLOPT - *COBYLA*	1. Rigid 2. Trefftz plane
STAR-CCM+	CFD Euler/RANS	-	1. Rigid 2. Surface-pressure integration

Panel lift forces are first obtained from an aerodynamic method, the doublet lattice method (DLM) - based aeroelastic solver in MSC.NASTRAN in this case. From the calculated strip-wise lift F_i, the strip circulation in the i^{th} strip is obtained through the Kutta–Joukowski equation as

$$\Gamma_i = \frac{F_i}{\rho V_0} \tag{59}$$

The total downwash on a strip i due to the circulation at all span-wise strips of the lifting surface is given by

$$w_i = \left(\frac{1}{4\pi}\right) \sum_{k=1}^{N} (\Gamma_k - \Gamma_{k+1}) \left[\frac{1}{y_i - y_k} - \frac{1}{y_i + y_k}\right] \qquad i, k = 1, ...N \tag{60}$$

The coordinate y_i is calculated at the centre of the strip (where the lift and circulation are calculated) and y_k at the outer definition of the strip (where the trailing line vortex is modelled) as shown in Figure 184.

The induced angle of attack in strip i is calculated knowing the strip downwash as

$$\alpha_{i_{induced}} = -\frac{1}{V_0} w_i \tag{61}$$

The induced drag in the strip i is finally calculated as

$$D_i = [F_i]\alpha_{i_{induced}} \tag{62}$$

where F_i is the lift force in the i^{th} strip and V_0 is the free-stream velocity. The definitions of the various coordinates are shown in Figure 184. The total induced drag is obtained by summing the contributions from the N strips on the lifting surface.

$$D = \sum_{i=1}^{N} D_i \tag{63}$$

It is to be emphasized that the D and D_i in the above equations refer only to the induced drag, that is, the drag due to lift.

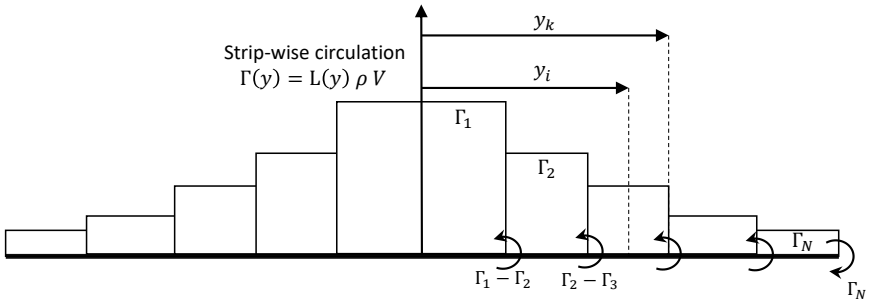


Figure 184: Definition of coordinates for the Trefftz plane implementation

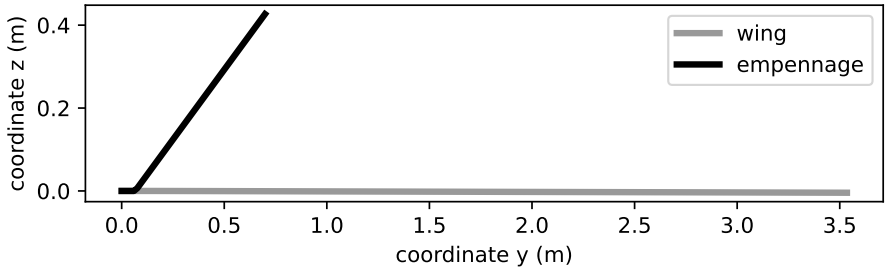


Figure 185: Wake at Trefftz plane from wing and empennage

Equation 60 can be used for a single lifting surface aligned along the horizontal plane as shown in Figure 184. In this case, the wake of this single surface is projected onto the downstream Trefftz plane. When two or more lifting surfaces are present, the influence of the wake of each lifting surface must be calculated on every other lifting surface's wake. Additionally, in the presence of a dihedral, the distance terms y_i and y_k in Equation 60 are replaced by the vector distance between the various interacting lifting surfaces.

The entire routine 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, that is Equations 59 - 63. 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.

VLM-based near-field implementation
The VLM-based near-field implementation presented by Kier [49] and summarized here was studied as a candidate tool in this exercise.

In conventional commercial aeroelastic codes for loads type application, a matrix based aerodynamics with a focus on the z-forces is commonly observed. The work presented in [49] extends the classical VLM implementation in the loads environment VarLoads [37] to also include induced drag by accounting for in-plane forces. The implementation of the VLM accounts for the inherently nonlinear behavior of the induced drag and the dependence on the on-flow direction, while preserving the aerodynamic influence coefficients and boundary conditions in matrix form, compatible with classical aeroelastic formulations. This is achieved by replacing the scalar product in the Kutta-Joukowsky theorem's classical implementation

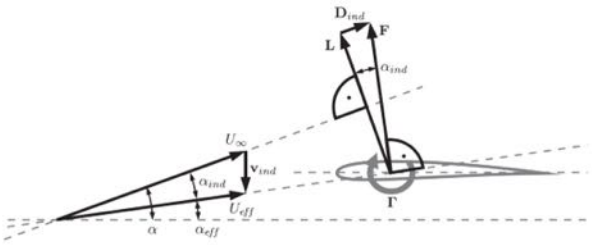


Figure 186: Induced drag and induced angle of attack [49]

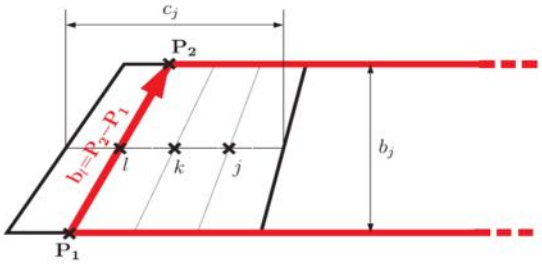


Figure 187: Geometry of aerodynamic box in VLM [49]

$$L_j = \rho U_\infty \Gamma_j b_j \tag{64}$$

with a vector product

$$\mathbf{L}_l = \rho \mathbf{V}_l \times (\mathbf{b}_l \Gamma_j) \tag{65}$$

where l and j denote the quarter chord and three quarter chord point of the aerodynamic panel box respectively as shown in Figure 187.

The difference with respect to a far-field approach such as with the Trefftz plane is that the induced downwash is calculated at the quarter-chord point of each box which turns the local lift vectors as shown in Figure 186. The calculated force includes the drag components and is beneficial because the approach provides a distributed drag modelling across the wing, implementable in flexible aircraft equations of motion, as opposed to the Trefftz plane which provides an integrated induced drag evaluation across each strip. Details of the actual implementation are provided in [49].

The optimization of the control surface allocation is performed in this case in MATLAB using the *fmincon* routine.

Drag estimation using the PANUKL code
PANUKL is a software package to compute the aerodynamic characteristics of an aircraft using low order panel methods [30]. 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.

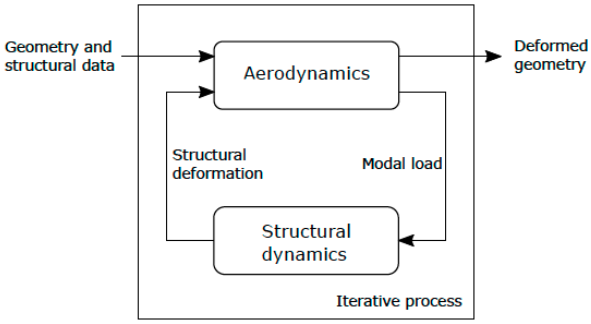


Figure 188: Trim flight deformation calculation process

- 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 as shown in Figure 454.

A key difference between the PANUKL solver and the DLM/VLM implementations discussed earlier is that the aerodynamic model in PANUKL is a 3D panel model. Camber and structural jig-twist are accounted for in the panel geometry. This isn't the case for classical DLM/VLM implementations which assume a flat panel aerodynamic model. In the case of the DLM/VLM tools discussed above, a downwash correction to account for the camber and jig-twist is included. Details of this PANUKL implementation are presented in [79].

AVL
AVL is a program for performing aerodynamic analysis of rigid aircraft of arbitrary configurations [21]. It uses the VLM to model the lifting surfaces. The T-FLEX UAV modelled with AVL is shown in Figure 189. One of the capabilities of AVL is the implementation of the slender body theory for fuselage modelling. Because of an intrinsic limitation of the VLM implementation, AVL is only suitable for inviscid calculation at small angles of attack and sideslip. For induced drag estimation, both surface pressure integration and Trefftz-plane calculation are available in AVL. In this study, the *COBYLA* optimization algorithm within the NLOPT Python package was wrapped around AVL to minimize the induced drag with optimal flap scheduling. Similar potential-flow solvers such as Tornado, PyTornado, XFLR5, VSPAERO, PAWAT and FlightStream were also tested and a comparison of these methods is presented in [122]. The results are also compared with Simcenter STAR-CCM+, a multiphysics CFD software. An important aspect of these tools is that they consider the aerodynamics to correspond to a rigid structure, as opposed to the tools in the previous sections which consider aeroelastic formulations.

Star-CCM+
In this study, the multiphysics CFD software Simcenter STAR-CCM+ was used to provide a comparison basis as a higher-order method. Given that the inviscid drag extracted from STAR-CCM+ is the pressure drag component acting on the aircraft, additional methods are required to extract the required drag components. These methods were not implemented at the time of writing of the paper and hence the results from the tool have not been elaborated further upon. Details of the CFD model and simulations performed are presented in [122].

Design of the new advanced FliPASED wing (TUM, DLR, SZTAKI)

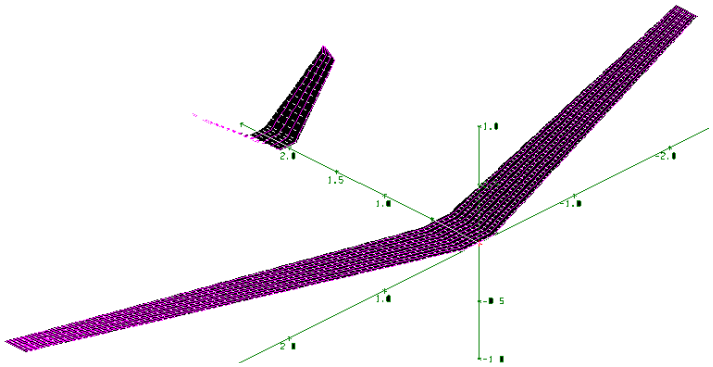


Figure 189: T-Flex aerodynamic panel model in AVL

According to the initial plan, the new aero-servo-elastic wing (-3 Wing) was to be designed using an established RCE toolchain. Given time constraints for the manufacturing and flight testing of new wing, the consortium decided to retrofit an old wing to decouple the MDO progress from the manufacturing and flight testing.

A feasibility check was conducted to make sure the retrofit plan was viable from mechanical aspects. The retrofit wing was designed to demonstrate the capabilities of maneuver load alleviation, gust load alleviation and wing shape control for drag reduction.

The 'reference' or -0 wings and 'tailored' or -2 wings from the Flutter Free FLight Envelope eXpansion for ecOnomical Performance improvement (FLEXOP) project were considered for the retrofit design study. A selection between the two wings was to be made based on the higher potential of demonstrating induced drag reduction during cruise flight. Additionally, the existing 4 control surfaces on the chosen wing were to be replaced by a larger number of control surfaces, aimed at demonstrating better drag reduction. The design of the two wing pairs is presented in [60, 72] and is the outcome of an aeroelastic tailoring design toolchain used within the project. The term reference here denotes that the wing was designed using conventional industry-near balanced symmetric laminates as against its counterpart, the 'tailored' wing which demonstrated higher passive load alleviation through composite tailoring.

The planform of the wing is shown in Figure 190. The existing control surface layout consists of 4 equally-spaced control surfaces starting from 12% upto 98% of the wing semi-span.

For this study, each of the existing four control surfaces on the wings were split into four control surfaces, that is 16 in total, in the simulation model. The drag minimization problem was solved for the reference and tailored wings at 1g trimmed flight for different flight points. The improvement in induced drag compared to the clean configuration (control surfaces at 0° deflection) attainable, for each of the respective wings is shown in Table 21. From this study, the following key inferences can be made.

- At the design speed of $45 m/s$, both wing pairs show a relatively less benefit of using active control for induced drag reduction. This is because the aerodynamic characteristics of the wings developed earlier during the project were aimed at this design speed. Consequently, the potential of drag reduction using active control is less at these flight points.
- In both wing pairs, the drag reduction potential is more prominent at flight speeds exceeding the design speed, where the wing deformation is larger and hence the potential for better wing shape control.
- The reference or -0 wing shows a larger potential at demonstrating drag reduction when compared to the tailored or -2 wing. The tailored wing was designed to demonstrate passive load alleviation

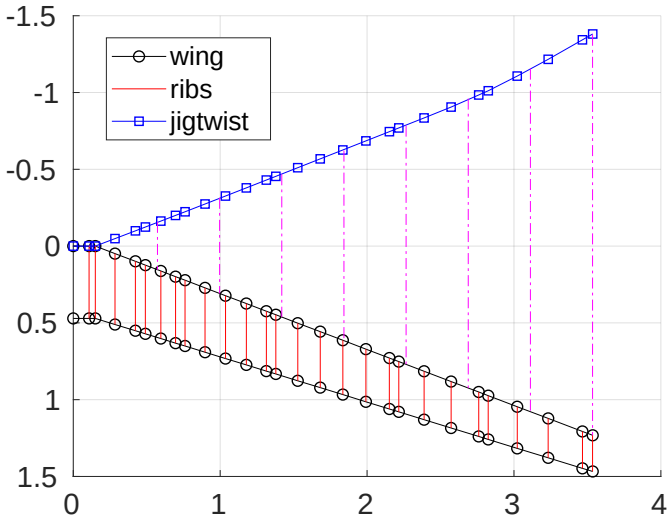


Figure 190: Planform of the reference wing used in this study

through composite tailoring. The inherent tendency of the wing to induce washout and shift loads towards the root results in a lift distribution that favors lower induced drag.

The lift distribution with and without active control together with the optimal control surface allocation at $50m/s$ is shown in Figures 191 and 192. The optimal control surface deflections corresponding to this flight point in the case of the reference wing are visualized in Figure 193.

Table 21: Comparison of induced drag improvement between reference and tailored wings (induced drag improvement is defined between clean and optimal control surface deflections of the respective wing)

	$20m/s$	$30m/s$	$40m/s$	$45m/s$	$50m/s$	$60m/s$
reference (-0) wings	3.6%	2.5%	4.3%	6.7%	9.9%	17.3%
tailored (-2) wings	4.8%	4.6%	4.6%	4.9%	5.5%	7.6%

Consequently, the choice was made to use the reference wing for the flight tests. In order to further increase the visibility of a reduction in drag through active control, design studies on modifying the deformation of the wing such that the wing is further away from its design point could be considered, for instance by adding a suitable mass at a favorable position of the wing, favorable in the sense of increasing demonstrability of drag reduction, which is the focus of the exercise.

The optimal control surface allocation for the reference wing for minimal induced drag is shown in Figure 191. As mentioned earlier, the simulation here assumes that each of the four existing control surfaces is split into four independent control surfaces. It is seen that close to the root and the mid-span of the wing, the variation in control surface deflection is quite low. The strongest gradients in the deflection are visible at the outboard sections of the wing.

An efficient solution would be to define smaller discrete control surfaces near the tip section where differences between adjacent control surface deflections are largest. Consequently, after studying different combinations, a layout where the existing four control surfaces from the root to tip are split into 1, 2, 3, 3 control surfaces respectively was chosen. This was done considering the drag reduction potential on the one hand and manufacturing feasibility on the other, keeping in mind the existing hardware and systems on the wing. For more details about manufacturing please refer to the deliverable [57].

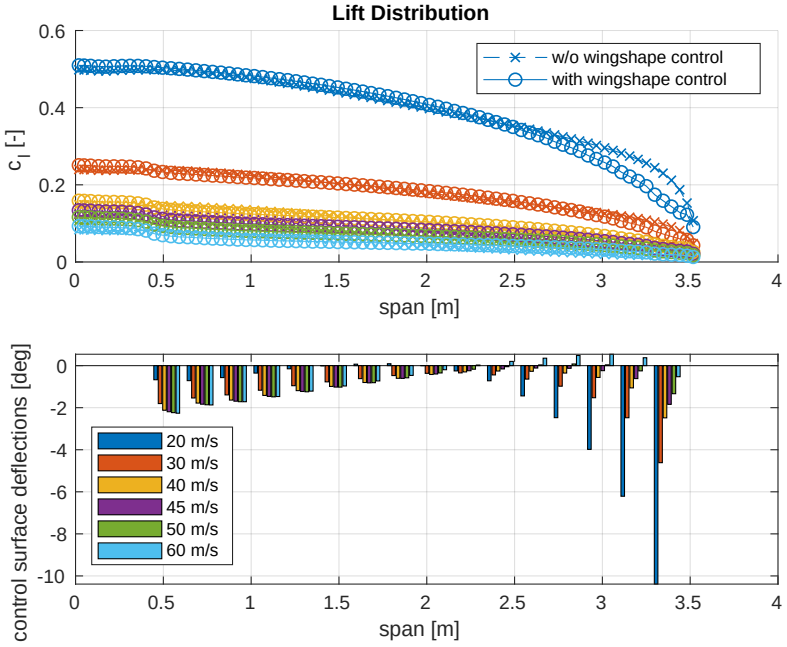


Figure 191: Lift distribution for the -0 reference wing without and with active control (above), optimal span-wise control surface deflections for minimum induced drag (below)

Analytical redundancy methods (SZTAKI)

Fault Detection

The goal of this section is to explore analytical redundancy methods for fault tolerance of the FLiPASED aircraft providing Fault Detection and Isolation (FDI) and sensor-actuator selection. However, as the aircraft is designed to test flexible wings with active control for flutter suppression and it is propelled with liquid fuel the effects of flexibility and the continuously changing mass can not be neglected. Thus first, the effect of flexibility on actuator and sensor fault detection design is examined and then the effect of the changing mass in gust load alleviation control design is explored.

One of the challenges in designing a FDI system for a flexible aircraft is to obtain an appropriate flexible model of it as opposed to rigid aircraft where modelling (or identification) is more traditional. Such a model is in general more complex and its construction requires special expertise. The report demonstrates that fast and accurate FDI for the FLiPASED aircraft indeed necessitates the use of a flexible model but if the performance criteria can be relaxed and the sensor configuration can be changed, a rigid aircraft model can also be sufficient.

For the flexible aircraft of the FLiPASED project, we want to detect two faults in the longitudinal motion of the aircraft: angle of attack sensor and elevator actuator faults. (Note that the tail of the aircraft is outfitted with ruddervators, therefore it would be more precise to say that we want to detect a fault in the ruddervators that affect the longitudinal motion of the aircraft. We will continue to refer to the control surface as elevator for simplicity.) The block diagram of the FDI filter design problem is depicted in Figure 194. We design optimal FDI filters with different bandwidths using the rigid and the flexible model of the aircraft. Then, using a simple decision mechanism, we calculate the smallest detectable fault and

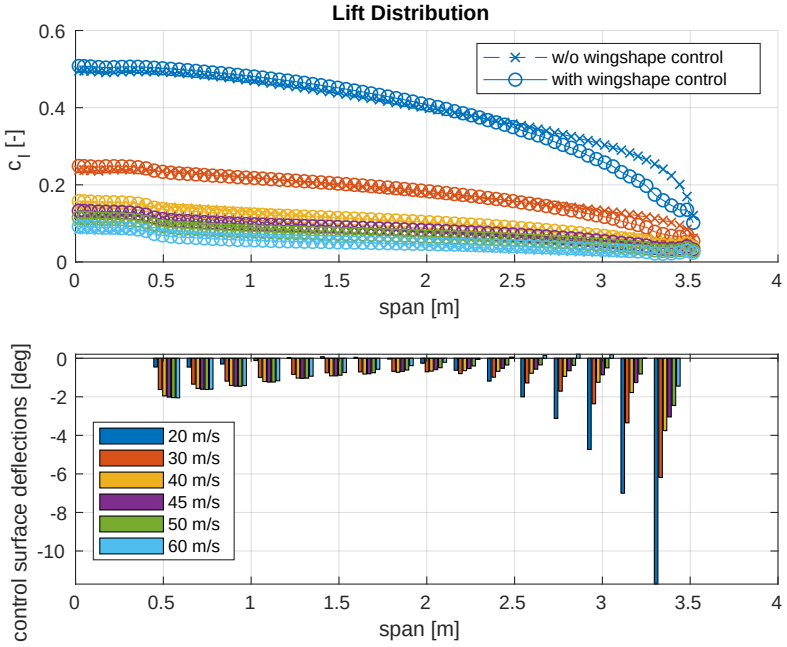


Figure 192: Lift distribution for the -2 tailored wing without and with active control (above), optimal span-wise control surface deflections for minimum induced drag (below)

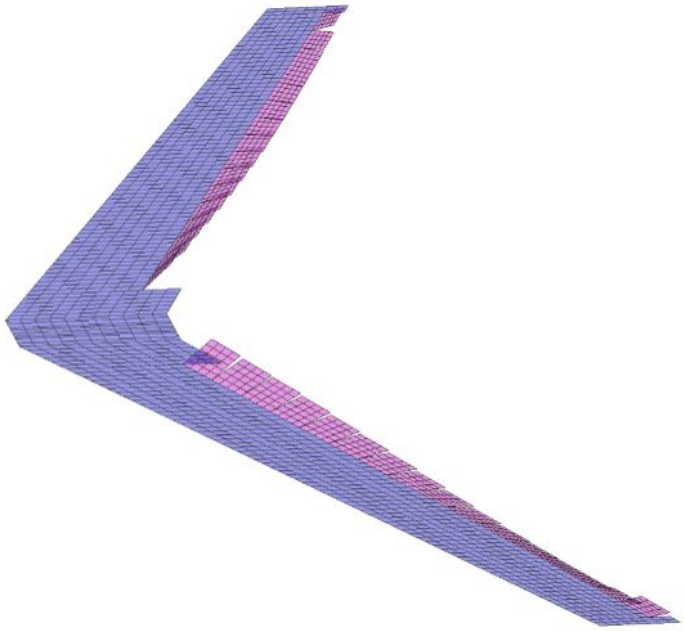


Figure 193: Optimal control surface deflections for the reference -0 wing at $50 m/s$ cruise (20x exaggerated control deflections)

the detection time for each fault and for each filter. Based on these results, we make recommendations

on what sensor configuration and which model to use for certain performance requirements.

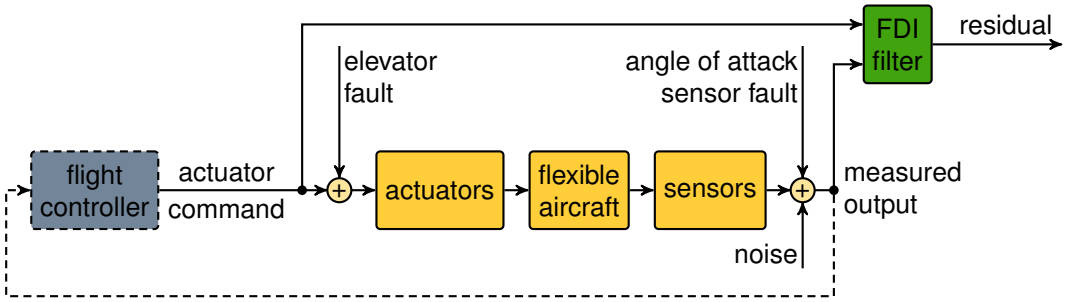


Figure 194: Block diagram of the joint actuator and sensor fault detection problem.

The sensors and actuators relevant for the fault detection are illustrated in Figure 195. Two models of this aircraft are used for filter design in this chapter: a low order rigid body and a higher order flexible model. Both are linear longitudinal models obtained in straight and level flight (at 38m/s). A detailed description is given by [105] and [71].

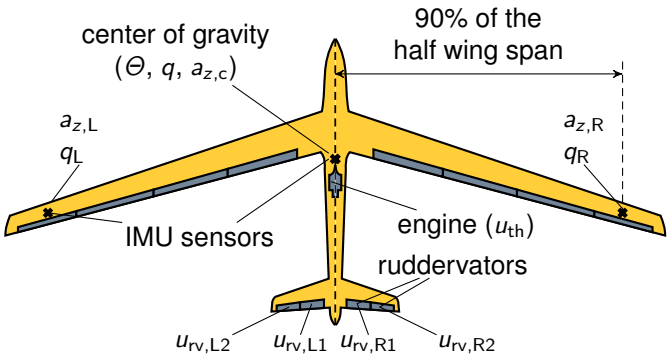


Figure 195: Control surface configuration and sensor positions of the flexible aircraft. The control inputs and sensor signals are marked at the corresponding control surfaces and sensors.

The outputs are the sensor signals that consist of the angle of attack (α), pitch angle (Θ), pitch rate (q), speed (V), vertical acceleration in the centre of gravity ($a_{z,\mathrm{c}}$), and the mean of the acceleration and angular rate signals from the IMU's located close to the wing tips ($a_{z,\mathrm{w}} = (a_{z,\mathrm{L}} + a_{z,\mathrm{R}})/2$, $q_{\mathrm{w}} = (q_{\mathrm{L}} + q_{\mathrm{R}})/2$, the 'w' stands for 'wing'). The sensors are modelled as first order low pass filters of the form

$$G_{\mathrm{sens}}(s) = \frac{1}{\frac{s}{2\pi\theta} + 1}, \tag{66}$$

where θ is the bandwidth. Additive white noise is assumed on the sensor outputs. Based on the documentation of the sensors and experimental data, the standard deviations of the sensor noises along with the bandwidths are listed in Table 22.

The thrust command for the engine is denoted by u_{th}. The tail control surfaces are ruddervators with the commands $u_{\mathrm{rv,L1}}$, $u_{\mathrm{rv,L2}}$, $u_{\mathrm{rv,R1}}$, and $u_{\mathrm{rv,R1}}$ in Figure 195. These are used symmetrically, i.e. $u_{\mathrm{rv,L1}} = u_{\mathrm{rv,L2}}$ and $u_{\mathrm{rv,R1}} = u_{\mathrm{rv,R2}}$. The elevator command considered in this chapter is obtained by

$$u_{\mathrm{e}} = \frac{u_{\mathrm{rv,L1}} + u_{\mathrm{rv,R1}}}{2} = \frac{u_{\mathrm{rv,L2}} + u_{\mathrm{rv,R2}}}{2}. \tag{67}$$

Table 22: Sensor bandwidth and standard deviation of the measurement noise.

	$a_{z,c}$	q	Θ
type	MTI-G-710 xSense		
bandwidth (θ)	200Hz		
std. dev. of the noise	0.08m/s^2	$0.3^\circ/s$	$0.6^\circ/s$
	V	α	
type	micro Air Data System 2.0		
bandwidth (θ)	50Hz		
std. dev. of the noise	0.33m/s	$0.33^\circ/s$	
	$a_{z,w}$	q_w	
type	MPU-9250		
bandwidth (θ)	200Hz		
std. dev. of the noise	0.72m/s^2	$5.4^\circ/s$	

Thus, the input of the system is the control command $u_c = \begin{bmatrix} u_e & u_{th} \end{bmatrix}^T$. Based on experiments, the engine dynamics can be approximated by

$$G_{act,th}(s) = \frac{1}{8s+1}. \tag{68}$$

The actuator dynamics for the elevator (for the ruddervators) is

$$G_{act,e}(s) = \frac{1817}{s^2 + 54.03s + 1817}. \tag{69}$$

Since the ruddervators are transformed to a single elevator, only one actuator is included in the model. The input of the aerodynamics consists of the control surface deflection, its derivative and second derivative, hence the derivatives of the output of $G_{act,e}(s)$ are also connected to the system.

The state of the system consist of the velocity components along the longitudinal and vertical axis of the body frame (u and w respectively), pitch angle (Θ), pitch rate (q), five modal coordinates and their derivatives, two lag states, and three actuator states. The frequency of the short period mode and the first bending mode of the structural dynamics have special significance in the final analysis. These are $\omega_{sp} = 9$rad/s and $\omega_{fb} = 18$rad/s, respectively.

The rigid aircraft model is obtained by residualising the flexible states (modal coordinates, their derivatives, and the lag states). In practice, a rigid model is usually the result of parameter identification of a standard rigid model. Our approach aims to avoid any differences between the two models that do not arise from flexibility.

The FDI filter design is articulated as an H_∞ optimal synthesis problem similarly to the solution of [67]. The generalised plant interconnection is depicted in Figure 196. Here, $f = \begin{bmatrix} f_a & f_s \end{bmatrix}^T$ is the fault which is modelled as an additive disturbance on the elevator actuator command and the angle of attack measurement. The output of the FDI filter $F(s)$ is called the residual. It is the estimate of the fault signal hence it is denoted by $\hat{f} = \begin{bmatrix} \hat{f}_a & \hat{f}_s \end{bmatrix}^T$. The control command u_c is normally the output of the flight controller but since no controller is considered in the design process, it is treated as a known external disturbance. The weighting functions are depicted in Figure 197.

Using a specific case study, guidelines are established on when a flexible model is required for FDI filter design for a flexible aircraft. It is concluded that only minor performance improvement is attainable for the angle of attack sensor FDI with the involvement of the flexible model. In contrast, the elevator FDI is greatly impacted by the choice of sensor configuration and design model. If good performance is

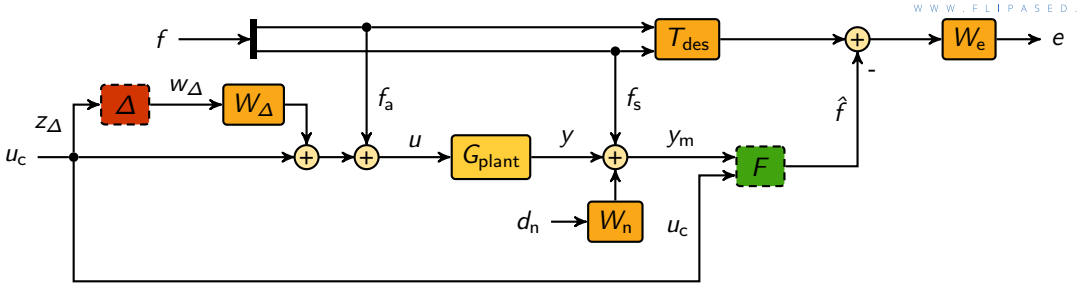


Figure 196: Generalized plant interconnection for the H_∞ FDI filter design.

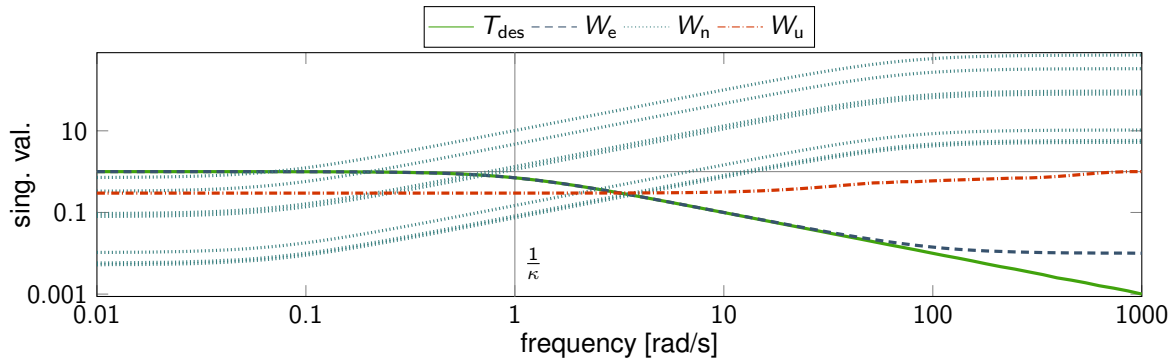


Figure 197: Weighting functions used for the H_∞ synthesis. The value of the design parameter is $\kappa = 1$s. (Since the standard deviations of the noise channels are different, $W_n(s)$ is represented by multiple lines.)

expected at high frequencies (beyond the frequency of the first bending mode), then both acceleration measurement at the center of gravity and the flexible model are required. Still using the acceleration measurement, good performance is achieved using the rigid model up to half of the frequency of the short period mode. At the cost of some loss in accuracy, a design based on the rigid model is capable of providing acceptable performance up the frequency of the first bending mode if the acceleration measurement is not used.

Gust load alleviation controller

Within the FliPASED project a multiple-model adaptive gust load alleviation (GLA) control system for the demonstrator aircraft is discussed and synthesised to solve the issue of increased vulnerability of modern aircraft configurations to gust encounters. Multiple-model adaptive control allows to identify the controller suiting best the aircraft's current properties like mass distribution by means of model detection methods. Different mass cases of the FliPASED demonstrator aircraft are considered by artificially attaching masses to the structural model. For each mass case a gust load alleviation controller is synthesized. Thus, a new control design approach is presented and applied to the demonstrator aircraft.

Validation of data science based methods for modelling and control (SZTAKI)

This section addresses the aspects of linear (parametrized) model approximation of dynamical systems, in view of control design. The model-free, or data-based approaches and their application to the flight data specific objectives will be described within the deliverable. In this work we are adopting big-data techniques to analyze the vast data provided by the complex sensing and control system. These methodologies are useful in mapping and revealing the underlying structure of the problem. Data

science technologies for optimal usage of these data are developed in FliPASED, and recommendations for methods and useful sensor arrangements for future aerospace applications are described.

The machine learning based approach results are presented through a flexible state estimation of the wings of the T-Flex aircraft. The investigated methods are described along with the used state-space model of the aircraft. The obtained results are presented and evaluated. Finally, conclusions are drawn.

The dynamic behaviour, stability, and the effects of the aerodynamic drag of a large-wingspan aircraft are mainly influenced by the structural flexibility and shape of its wings during flight. Large commercial aircraft has large mass variation during flight, as fuel is consumed, hence optimal (minimum drag) configuration at one point of the mission might not be optimal in other parts of the flight. Aircraft design accounts for this change by simultaneously optimising the wing lift and drag for multiple points within the flight, but the typical optimization relies on passive means with the assumption that flaps have to be at zero deflection during the trimmed cruise phase of flight. On the other hand if a database (most likely derived by CFD tools) is available about the optimal wing shape and the corresponding flap deflections, leading to minimum drag at each point within the cruise flight envelope, significant reduction can be achieved in terms of fuel consumption. For each individual point in the flight envelope the optimal wingshape has to be achieved by an adequate wingshape controller, what might not only contain flap scheduling but also setpoint tracking of the optimal modal coordinates of the wing. For estimating the modal coordinates and reconstructing the wing shape a state observer is necessary because the direct and accurate measurement of these states is not feasible. Two approaches are investigated as possible solutions for the state estimation task. First, Extended Kalman Filtering (EKF) is presented, using a Linear Parameter Varying (LPV) system model. Second, a machine learning-based approach is introduced based on the new KalmanNet architecture with two different recurrent neural network configurations: one with linear layers and one with one-dimensional convolutional layers.

Extended Kalman Filtering (EKF) is used as a model-based wing shape estimation approach. The EKF is the extension of the standard Kalman filter to be used with nonlinear systems for state estimation and sensor fusion. The EKF pipeline requires the full, nonlinear state-space description of the system and information about the model noise and observation noise in the form of noise covariance matrices (denoted as Q and R respectively).

The other approach for estimating the flexible states of the T-Flex is to use machine learning. We employ the recently published KalmanNet architecture [90]. The algorithm (or *pipeline*) for the KalmanNet is presented in Figure 198. KalmanNet combines Kalman filtering with a neural network as it uses similar *prediction* and *update* steps, but without computing the state prediction covariance matrix (P). Consequently, the model noise covariance matrix (Q) is not involved. For providing the Kalman gain (Step 4 in Figure 198), a trained Recurrent Neural Network (RNN) is used, thus the observation noise covariance matrix (R) is not involved either. The neural network uses the innovation difference $\Delta y[k] = y[k] - \hat{y}[k|k-1]$, the forward update difference $\Delta\hat{x}[k] = \hat{x}[k-1|k-1] - \hat{x}[k-1|k-2]$, and the roll angle ϕ scheduling parameter as input features. The advantage of the KalmanNet compared to the EKF is that it does not require any information about the model of the noise processes and the promise of better generalization capabilities.

The standard Kalman gain predicting neural network [90] uses a Gated Recurrent Unit (GRU) as the recurrent layer and linear layers with Rectified Linear Units (ReLU) as the activation function. The neural network has a linear layer as the input layer with ReLU activation, followed by the GRU. After the GRU layer, there is another linear layer with ReLU activation, then the linear output layer. As the aircraft model we use is high-dimensional (48 states, 64 outputs), we slightly decreased the dimensions of each layer compared to the original architecture to reduce the computation burden.

Apart from the *linear RNN* architecture, we implement a different neural network that still uses a GRU cell, but instead of linear layers, it uses three convolutional blocks at the beginning of the network [124]. A convolutional block consists of a 1D convolutional layer followed by a ReLU activation function. After

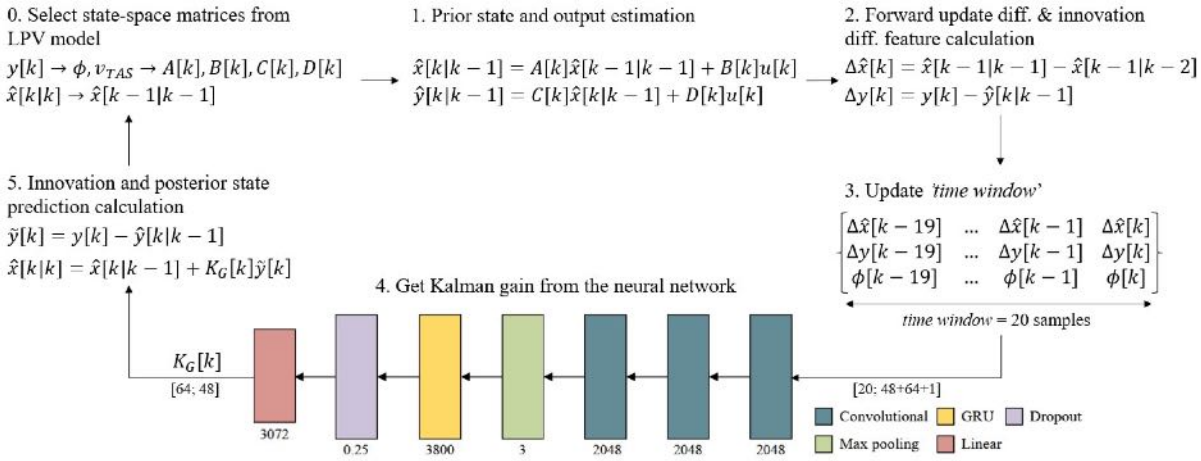


Figure 198: KalmanNet pipeline

the ReLU a Batch Normalization layer is used, followed by a Dropout layer with 0.25 dropout probability. The output layer is a linear layer, which provides the Kalman gain matrix. The kernel size for each 1D convolution layer is seven. As the 1D convolutional layer requires a trajectory, or time-window of input features, simply using the forward update difference ($\Delta y[k]$), innovation difference ($\Delta\hat{x}[k]$) and roll angle (ϕ) input features of the current time step is not adequate. Therefore, we use the input features of the current time step and the input features from the previous 19 time steps in the time-window buffer. In Figure 198, the architecture with the convolutional layers represents the neural network. The number of features is shown below the convolutional blocks and the pool size below the max pooling layer. The number of units is indicated underneath the GRU and the linear layer. The dropout rate is shown below the dropout layer.

From here on – for the sake of brevity – the original KalmanNet architecture is referred to as the *linear* RNN architecture, while the second, new architecture is referred to as the *convolutional* RNN architecture after their defining layer types.

For initializing the layer weights, a standard normal distribution is used. Since the architecture incorporates a discrete-time system, it has a high sensitivity to the initial weight values. Therefore, the standard deviation of the normal distribution for the initialization has to be chosen very small ($5 \cdot 10^{-6}$) to avoid the otherwise highly diverging training process.

For training a neural network, generally three different datasets are required: training, validation, and test datasets. The training dataset – as its name suggests – solely used for optimizing the weights and biases of the neural network. The validation set is used for testing the performance of the network during training on new data samples. The purpose of this is to monitor the stability of the training, to detect overfitting, and to fine tune hyperparameters. (Overfitting is the phenomenon when during training the network is no longer capable of getting lower loss values while maintaining its generalization capabilities and starts to memorize the training data, thus reaching smaller loss values on the training set, but greater and greater losses on previously unseen datapoints.) The test dataset is only used after the network is fully trained to obtain final performance metrics. The training, validation, and test datasets are generated using the high-fidelity nonlinear Simulink model of the T-Flex.

The training dataset has four different trajectories that are generated with the help of the baseline controller [84]. These four trajectories are the following:

- the oval-shaped 'horserace' track,

Architecture	Linear RNN	Convolutional RNN
Learning rate	$3.2 \cdot 10^{-6}$	$7.5 \cdot 10^{-5}$
Weight decay	$1.5 \cdot 10^{-7}$	$9.5 \cdot 10^{-5}$

Table 23: Hyperparameters

- an '8-shaped' track,
- a trajectory where the controller only receives roll angle (ϕ_{ref}) reference signals,
- a trajectory where the controller receives altitude (h_{ref}) and velocity (V_{ref}) reference signals.

These four trajectories are created with the intention to cover as many possible real-life flight conditions as we can in order to enhance the generalization capabilites of the neural networks. Also, to create rich datasets, while having realistic flight conditions, randomized wind gust and turbulence disturbances are used, together with Gaussian sensor noise, based on the flight test results of the T-Flex [6] for each dataset. The other purpose of applying wind loads is to have disturbances that cannot be incorporated into any covariance matrix. The trajectories of the training dataset are split into eight, 96-second long batches. The sampling time is set to 5ms, which results in 19200-sample long training batches. For training, a single batch is randomly selected from the eight in each epoch. Validation and testing are conducted using only a trajectory where the aircraft follows the '8-shaped' track. The initial velocity is set to 42m/s in all cases. The possible range of airspeed changes is between 39m/s and 51m/s, for the roll angle between $0°$ and $45°$. The barometric altitude can change between 780m and 820m.

It is important to mention that the performance of the linear RNN architecture proved to be more stable than the convolutional RNN, which tends to get stuck in local optima. So, to overcome this issue, a reduction of the learning rate during training (called learning rate scheduling) is necessary in that case. The threshold is set at -21dB – according to the decibel-based error metric – and the reduction factor is 0.05. The new learning rate is calculated as $\mathrm{lr}^{\mathrm{new}} = \mathrm{factor} \cdot \mathrm{lr}^{\mathrm{old}}$.

The performance of the different methods and architectures are evaluated on the 96-second long test dataset, where the aircraft follows the '8-shaped' track with wind and turbulence disturbances present. Since the main purpose of the state estimator design is to observe the states describing the flexible dynamics, only the results for these states are presented.

In Table 24 the prediction errors are presented in the RMSE metric used throughout the training of the neural networks. Based on this metric, the performance of the two learning-based approaches are better than the LPV-based EKF: having only half the total error value. When looking at the three different state groups (modal coordinates, derivatives of the modal coordinates, aerodynamic lag states) the followings can be observed. When estimating the modal coordinates, all three architectures provide similar performance. For the lag states the LPV-based EKF provides better performance than either architecture. In the case of the derivative states, where the accurate estimation of the states is a rather challenging task because of the significant level of disturbance, the learning-based methods fare much better than the LPV-based EKF. However, as discussed during the time domain analysis of results, this does not mean that any of the architectures can completely negate the effects of the disturbances. Comparing the two neural network architecture the linear RNN has slightly better performance as metrics concerned. However, it took significantly less time to train the convolutional RNN while it also used less GPU memory and the size of trained model is smaller than for the linear RNN.

Architecture	LPV-EKF	Linear RNN	Convolutional RNN
Total RMSE	0.0120	0.0053	0.0066
U_{fn} RMSE	$5.91 \cdot 10^{-4}$	$8.82 \cdot 10^{-4}$	$9.79 \cdot 10^{-4}$
$\dot{U}_{fn}$ RMSE	0.0183	0.0080	0.0100
lag_n RMSE	$6.95 \cdot 10^{-4}$	0.0014	0.0014

Table 24: Prediction errors

2.2.4 Deviations, their reason, impact on the project and corrective actions

The main issue encountered is the numerical ill-conditioning of the reduced order models, especially in the case of the baseline control design block. This caused delays in the tool adaptation. However, this delay does not affect the other control design blocks (flutter, GLA, MLA). Improving the numerical conditioning of the low order model is currently investigated and the LTI algorithms for the baseline control design result in feasible controllers. Further investigation is required for the gain scheduled design.

2.3 Explanation of the work carried out per WP - Work Package 3

2.3.1 Objectives and activities

The Work Package 3, Demonstration and Testing, has the following objectives:

1. Model refinement using GVT data
2. Model refinement using flight tests
3. Performance verification of active control methods

In addition, the activities, related to all mechanical work such as manufacturing and integration are also covered by the work package.

Task 3.1, Demonstrator Baseline, has seen much activity starting in December 2019, with planning of the needed upgrades for safe operation of the demonstrator. After a crash of the T-FLEX demonstrator, additional work had to be performed to get P-FLEX flying.

For the Task 3.2, Demonstrator Wing Design, a sensor concept has been discussed for the new wing design, as well as an alternative plan. Manufacturing a completely new wing became impossible due to time constraints.

Task 3.3, Manufacturing and Integration, had activities related to design and manufacturing of a new control module of the Flight Control Computer, the RX-MUX-II. Also, a complete redevelopment of the communication between FCC and the RC system has been done.

A lot of the work has been performed under Task 3.4, Ground Testing of the Demonstrator. This included software updates and integration, multiple taxi tests of the upgraded landing gear and simulator training in preparation for the flight tests with T-FLEX After the build of P-FLEX, a successful GVT campaign was performed. This was followed by several system ground tests, to get the new aircraft flight worthy.

During Task 3.5, Flight Test Specification and System Identification, plans for a total of four flight test campaign have been made. One test flight took place in 2021, where important milestones have been achieved regarding the landing gear and also sensor updates. Due to changed regulations, the following flight test campaigns took place in Cochstedt (EDBC) instead of Oberpfaffenhofen (EDMO).

For fulfillment of Task 3.6, Flight Test Campaigns, three flights took place in 2021 and four campaigns took place in 2022 and 2023. With a total of 31 flights, the complete envelope was flown. Flutter suppression, using closed loop controllers, was demonstrated in the last three flights.

2.3.2 Starting point and approach

Initial state of the demonstrator The project for TUM has started with a demonstrator, which has already been used in the previous project, FLEXOP. The demonstrator (Figure 199) has performed six flight test up to then. However, building on previous experience, landing gear proved to be one of the biggest challenges during the operation of the demonstrator. The aircraft was very difficult to control while on the ground, leading to a few very dangerous situations and one accident, where the aircraft skidded of the runway and hit a runway light. Therefore, upgrades were necessary to ensure sustainable operation of the aircraft.

Figure 199: FLEXOP Demonstrator during the last flight previous year

As a starting point, the following design flaws have been identified:

1. The maximum angle of attack, achieved on the ground, is limited by very low main landing gear and a high tail wheel. This design solution limits the maximum angle of attack that could be achieved for takeoff to 3.3deg. This is very small for a taildragger aircraft and usually would be around 10deg. In addition, fixing such a design on an already manufactured aircraft is not easy.
2. Very narrow main landing gear makes it easy for the aircraft to bank from wingtip to wingtip. If this happens during takeoff or landing, the wingtip touches the ground and instantly creates a destabilizing moment.
3. Main landing gear is longitudinally far from the center of gravity. This means that the disturbing bank angle, required to tip the aircraft, is further decreased.
4. The tires of the main landing gear are too soft for the airplane. This makes it possible to deform the tires very easily and also significantly increases the rolling resistance during take-off run.

5. Unsteerable tail wheel makes the aircraft very hard to control while on the ground. The tail has to be lifted up first and aircraft is then steered with the rudder.
6. Retractable main landing gear proved to be an unnecessary design add-on to the aircraft which adds complexity, but not value to the demonstrator overall.

These problems were hard to identify during the conceptual or preliminary design phase of the FLEXOP project and were only realized during operations. Therefore further discussion was held how to make the controllability of the aircraft better (Figure 200) .

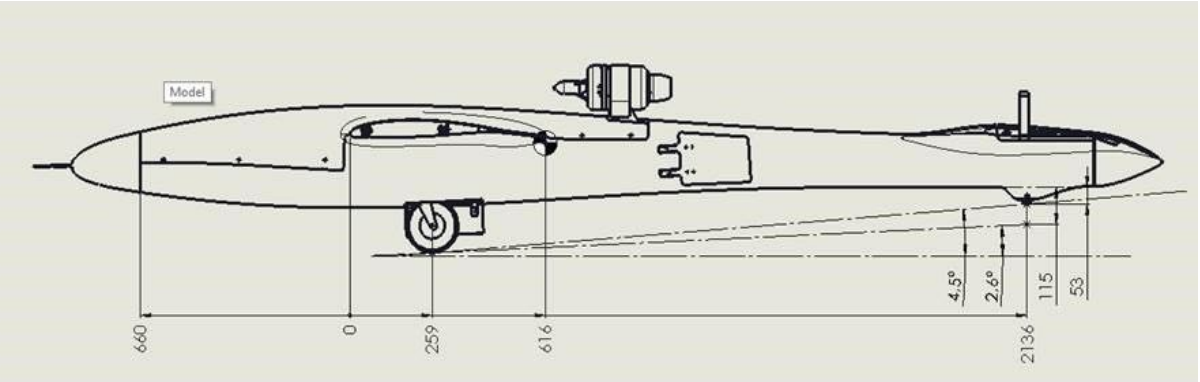


Figure 200: Comparison of the maximum angle of attack during take-off. 4.5 degrees is the initial tailstrike angle, 2.6 degrees is the tailstrike angle with steerable tailwheel assembly (wing incidence angle is -1.2 degrees)

Another objective during the first year of the project was to improve the operations of the demonstrator. This was done in three areas: streamline the operational procedures at the airport, change the electronic wiring to decrease number of actions required to set the aircraft up and improve role redundancy within the team. Therefore, further meetings were setup within the flight test team to discuss and streamline the preparation guidelines as well as think about how to make the crew planning easier. In addition, issues were identified in the electrical system of the aircraft that made the complexity of operations higher than it could be.

Since the data, gathered from flight test, had to be processed, some processing toolchains have already been implemented from before. Sensor errors were already being dealt with, as well as logging errors. The end product would be a single file with clear data structure inside that could be used with MATLAB for further analysis. However, the ultimate goal is to streamline the processing of the data as much as possible. This would include a completely automated data processing, where very minimal operator action is needed. In addition, the automated processing would compile a preliminary test report, allowing to analyse the outcome of the test on the fly.

2.3.3 Efforts and achieved results, name of involved partners

Wing -3 Advanced Wing In order to adapt a -0-wing for being suitable to measure the effect of actively shaping the lift distribution for the reduction of induced drag the number of available flaps on each wing has be increased from four to nine. The considerations behind following an approach to increase the number of flaps and distribute them accordingly are described in [48]. To achieve this goal, the following changes are planned:

- Replacing flaps on position 2, 3 and 4 with newly manufactured flaps.
- Outfitting the -3-wing with five additional servos to actuate the flaps.
- Adjusting the wiring to supply the additional servos with power and control signal

The required changes are implemented based on the existing CAD-model of the -0-wing in the software *CATIA V5*. The reuse of existing component models whenever appropriate increases the efficiency of design as well as reliability since the experience from the implementation of the -0-wing exists.

The design of the flaps follows a standard approach for later manufacturing using a manual lay-up process in a negative mold, based on the CAD-geometry developed during the predecessor project *FLEXOP*.
Each flap consists of a upper and a lower skin, a foam core, flap horns for actuation and a guiding tube for later insertion of hinge wires serving as an axle. The flaps are manufactured by using negative molds that provide the geometric outer surface for the production of the outer skins. After laying in the foam core and guiding tube, the two mold sides are stacked upon each other to form the closed flaps

For the fixation of flaps a pragmatic approach is chosen by using nut-plates to locally reinforce the rear spar of the -3-wing and eye-bolts to provide hinges that are adjustable both in depth as well as the axis, thus facilitating the hinge wires serving as an axle.

In order to select appropriate actuators, a study of hinge moments has been conducted using the freely available program *XFLR5* [20] aiming an determining the maximum expected hinge moments. The operating conditions for the simulation are subsequently chosen to the following, rather extreme, values in order to assure a sufficient safety margin. The final selection will be taken once the implementable wiring scheme is determined during the manufacturing process (please refer to the feasibility study described in deliverable D3.4). Due to similar form factors an exchange of servos is possible.

The biggest challenge is deemed to be the additional wiring, which already posed a challenge in the design of the -0-wing.

For installing the wires, routing of new cables is a preferred solution using a pragmatic approach supported by a keyhole camera. The new servo positions, that cannot be reached with new cables due to restrictions in space are supplied using existing cables and routing Y-cables. If no new cables can be routed, the existing CAN-cables in combination of CAN-utilizing servos will be used.

Two CAN buses have been installed earlier in each half wing, which are used by FlightHAT for reading IMU and SHM data on the top buses and IMU data on the bottom buses. Bottom IMUs placed as spare sensors and were not used. Therefore, they can be disconnect them from the bus without loss of system functionality and the bus-cable can be reused for servo control. The top buses can be connected for RX-MUX-II and FlightHAT in the same time and used for sensor reading and servo controlling as well. Figure 201 shows the wiring solution of the -3 wing. For safety reasons, the control surfaces are supplied in a way which resembles the setup of the original -0-wing, thus retaining the redundancy characteristics developed during the design of the *T-FLEX* flight demonstrator, i.e. a controllable aircraft, even if one of the RX-MUXes, batteries or other components such as an individual servo actuator has stopped working.

In conclusion, the implementation of the changes to adapt an existing -0-wing to create a -3-wing has

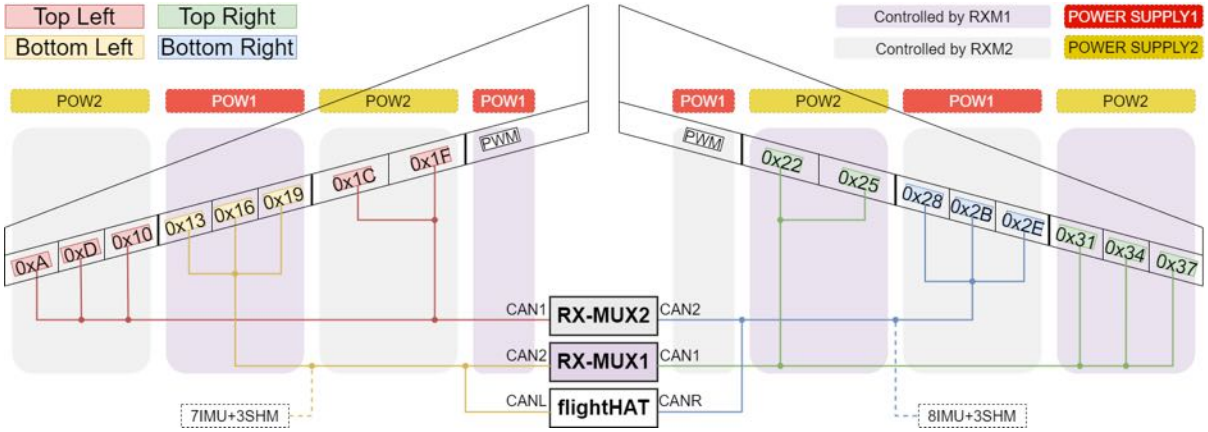


Figure 201: Wiring of -3 wing

been planned to rely on pragmatic technical approaches and techniques. By and large, the implementation of the changes is deemed technically feasible.
Currently, the described implementation has commenced according to the design described and the state can be described as follows: The molds are produced and the flaps are manufactured to sufficient quality in terms of geometric accuracy. The routing of wires has been investigated, implemented to large extend already and a decision in favour of CAN-utilizing servos has been taken. Required mechanical parts such as flap horns and gluing templates are being manufactured using a CNC-router.
The next steps will include the refinement of the existing CAD-design and testing of technical solutions such as adhesive bonding on mock-up test-stands, setup of sub-assemblies and finally the integration and functional testing of the wing assembly.

Aerodynamic investigations were done in order to assess the improvements of the aerodynamic (induced) drag coefficient $C_{D,i}$ when wing shape control techniques are applied. The goal of the investigation was the comparison of effects of aforementioned technique on the -0- and -2-wing. The following figures 202 shows the change in lift distribution over a range of airspeeds V for the -0-wing.

The improvements of the induced drag coefficient $C_{D,i}$ are shown in the following table 25

Wing	$20\frac{m}{s}$	$30\frac{m}{s}$	$40\frac{m}{s}$	$45\frac{m}{s}$	$50\frac{m}{s}$	$60\frac{m}{s}$
-0	3.59%	2.47%	4.32%	6.7%	6.87%	17.32%
-2	4.82%	6.62%	4.56%	4.88%	5.48%	7.58%

Table 25: Improvements of the induced drag coefficient $C_{D,i}$ at different airspeeds of the wing -0 and wing -2 due to wing shape control technique.

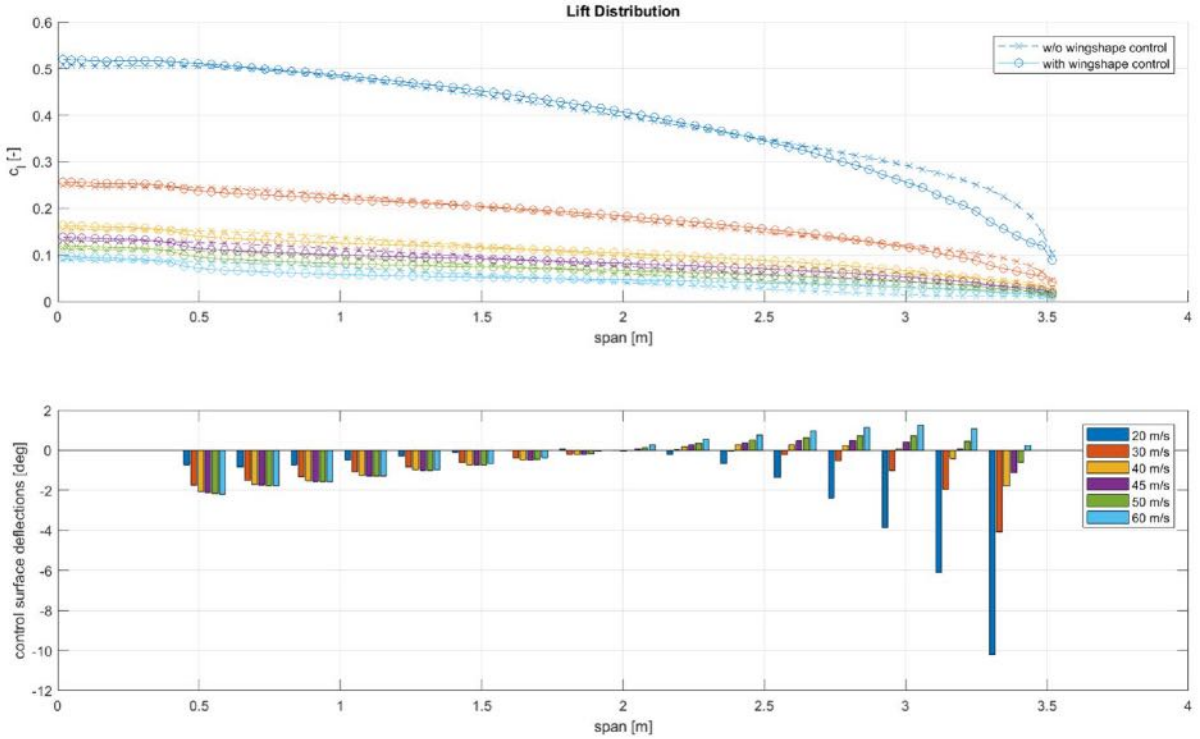


Figure 202: Wing -0: Lift distribution with and without wing shape control at different airspeeds and the respective flap deflections.

Improving the landing gear (TUM) Two different concepts for fixing the landing gear were discussed:

1. Fundamentally changing the landing gear layout.
2. Adjusting the current landing gear to make it acceptably safe for operation.

Because of the fact that the first option would require major fuselage changes and would take at least a few months, it was decided to start with the second option first. Ways to improve handling were discussed during the winter before the first flight test campaign. Due to the complex nature of the problem the solutions that were initially agreed upon did not completely resolve the issue. This resulted in an iterative process with different concepts being implemented as add-ons to the initial design along the way. The chronology of the process was:

1. Implement the steerable tailwheel with damping
   (a) The initial solution to steering was to install an off-the-shelf tailwheel assembly. Unfortunately, the solution did not work because the load on the tailwheel appeared to be too big for the part. Therefore another, completely custom iteration was done. This included a custom milled aluminum fork for steering and a damping assembly. The damping assembly was composed of glass-fiber-reinforced plastic plate acting as a leaf spring for longitudinal damping and two rubber dampers for lateral stiffness. The structure held well, but the steering made the aircraft hard to control and very sensitive to any pilot inputs.
2. Change the brakes of the main landing gear to more effective ones

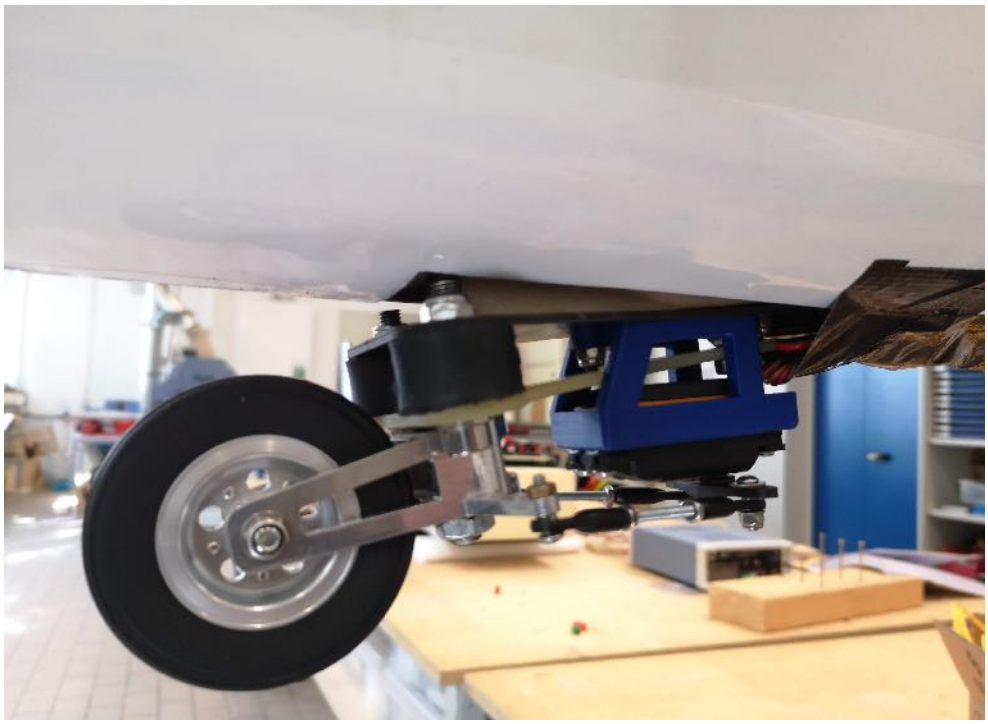


Figure 203: Steerable tailwheel assembly

(a) Tire brakes were changed to drum brakes. From previous testing it was noted that the tires wear out very quickly due to the brakes. Also, the braking power of the old system proved to be too little. Therefore, new type of brakes was implemented that would both conserve the tires and increase the braking force on the wheel hub.

3. Add a gyro to the tailwheel

   (a) Introducing the steerable tailwheel did not solve the controlability problem as the team has hopped. The aircraft became very sensitive, especially at higher speeds. The solution was to introduce a gyroscope-based compensation for the gain on the steering. This proved to improve the steering somewhat.

4. Reverse the main landing gear frame to shift the ground contact point back

   (a) One of the main findings, mentioned in the early research on taildragger aircraft is that the tendency to veer of the runway is decreased if the centre of gravity is kept as close as possible to the main landing gear. This was recorded in all the reports on the topic. Therefore, changing the location of the landing gear was considered. Luckily, the landing gear frame was easy to flip, moving the main landing gear backwards by 75mm. The outcome was lesser tendency to veer off the runway, an increase to the critical bank angle to tip on one wing, but also higher load on the main tires. Even though the weight increase was only 2.5% per wheel, the main tires were already overloaded before. The further steps would include looking for stiffer main tires, if possible.

5. Laterally stiffen the main landing gear assembly

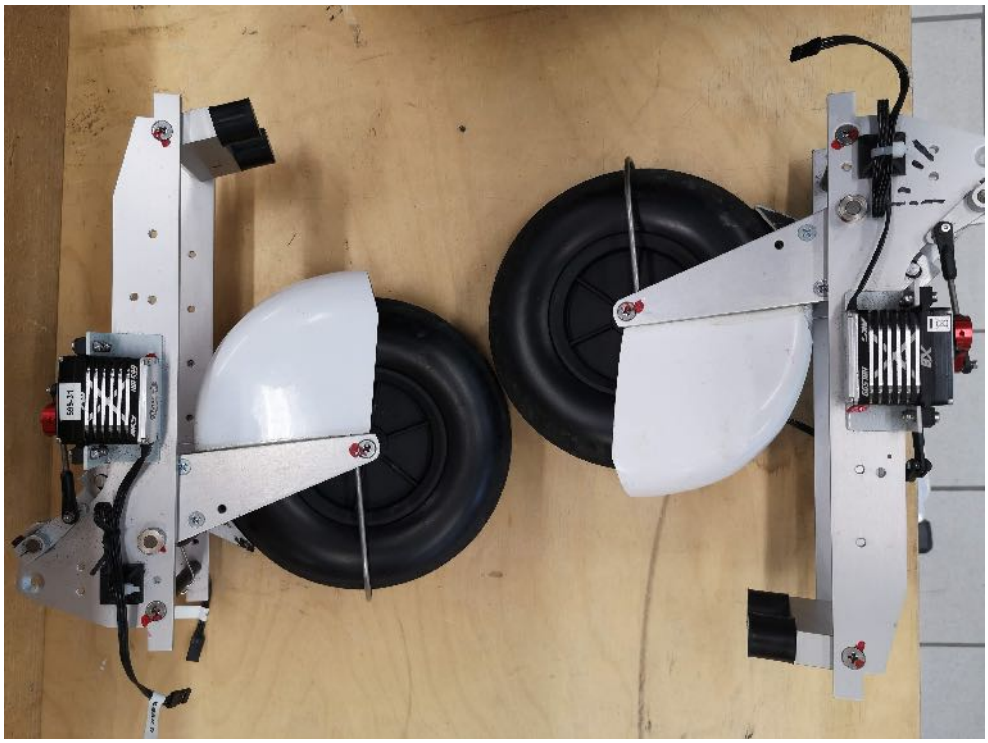


Figure 204: Comparison of two possible positions for the main landing gear. The difference is around 75mm

(a) During the taxi tests cameras were mounted facing both the gears. This helped to observe the behavior of the landing gear and make further conclusions. One of them was that the main landing gear is too flexible laterally, which makes it easier to tip onto one wing and harder to get out of the tipped position. Therefore, further parts were introduced to stiffen the landing gear laterally.

6. Change the main wheels to stiffer ones

   (a) Even though the gear was made stiffer, it was recognized that the tyres of the main gear are way too soft for the aircraft. This was discovered during one of the testing days, where the aircraft stood on the ground for a couple of hours. As a result the foam-filled tyres deformed plastically and were not usable anymore. Additionally, during high speed taxi tests a set of tyres burst into pieces after they got too hot (Due to braking and rolling). It was decided that a stiffer tyre is a must. And with no alternative tyres available for the same wheelset, a double sailplane tailwheel (TOST 150 MINI) instead of the original RC model grade wheels were bought. The TOST wheels would have a proper inflatable tyre mounted on, which would make the main gear stiffer laterally.

7. Add brakes with higher efficiency

   (a) In addition to upgrading the wheels to stiffer ones, the TOST wheels also had a possibility to have disc brakes mounted on them. Since long braking path was also discovered to be a problem during our flight tests, this seemed like a good option.

The changes of both, main gear and tailwheel resulted in a considerably more steerable aircraft. Multiple taxi tests were done, including low speed and high speed tests, to make sure the aircraft has

Figure 205: Too soft tires deforming under normal load

enough controllability to safely resume flight testing. In the end, changing the main wheels from RC model grade to aviation grade seemed to make the biggest difference. The aircraft was declared as flight-worthy again.

After the crash of the FLEXOP demonstrator, additional changes were performed during the rebuild process:

8. Remove retractable Landing gear

   (a) During the early stages of the rebuild process, discussion and investigation were made for the usefulness and the need for the retractable landing gear system. The main reason to have retractable landing gear was to decrease drag during the tight acceleration phase of the flutter tests. From experience with the previous system, and as well as with additional analysis, we found the following: Having the landing gear retractable, increases system complexity and weight. It is not possible to make it laterally stiff enough - given our present constraints. It is possible to increase 40%-50% fuel capacity, if the internal space would be used for fuel storage instead. In favor of extended fly-time, a fixed landing-gear system is designed and integrated to the fuselage.

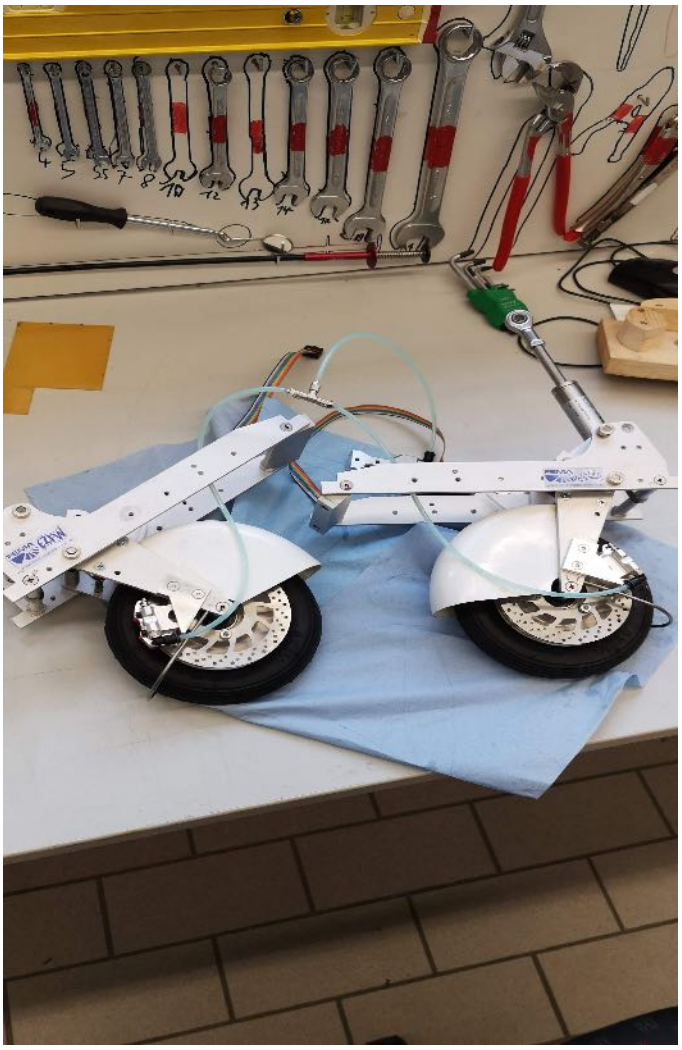


Figure 206: New main wheels being fitted with disc brakes before installation

Air brake/ During the early stages of the rebuild process, discussion and investigation were made for the usefulness and the need for the integrated air brake system.

Main reason to have an operational air brake:

- Use it for precise airspeed control
- Increase drag during landing

During the investigation, we found:

- Air brakes are not used for speed control
- During landing, fully opened air breaks are as effective as 10% decrease in throttle command.
- Circles will be flown instead of horse-race for flutter tests, so drastic deceleration and accelerations are not required

- If possible, the additional weigh could be reused for additional fuel, to increase flight-time.

Due to the reasons mentioned above, the air-brakes are not integrated into the fuselage, thus helping to decrees system complexity and increase weight capacity in favour for additional on-board fuel.

Fuel tank and Fuel Transfer system Investigation showed, that during a full circle flow, 0.14kg, 0.26kg, 0.41kg and 0.59kg fuel is used if the circle if flow with $30\frac{m}{s}$, $40\frac{m}{s}$, $50\frac{m}{s}$ or $60\frac{m}{s}$, respectively. A possible 40-50% fuel capacity increase will be highly beneficial to maximize the positive outcome a single flight.

Due to that, as it is already mentioned in 2.3.3 and in 2.3.3, an additional design element with the rebuild is to increase the possible fuel capacity. That is achieved mainly designing and manufacturing custom fuel tanks for the space, where the retracted landing gear is located in T-FLEX.

Using the space available, and construction a custom solid tank, it would be possible to increase fuel capacity by ∼47%. To aim that goal, a custom solid tank were build. Figure 207 shows one of the auxiliary fuel tank modules. The from the layout two is integrated into the fuselage.

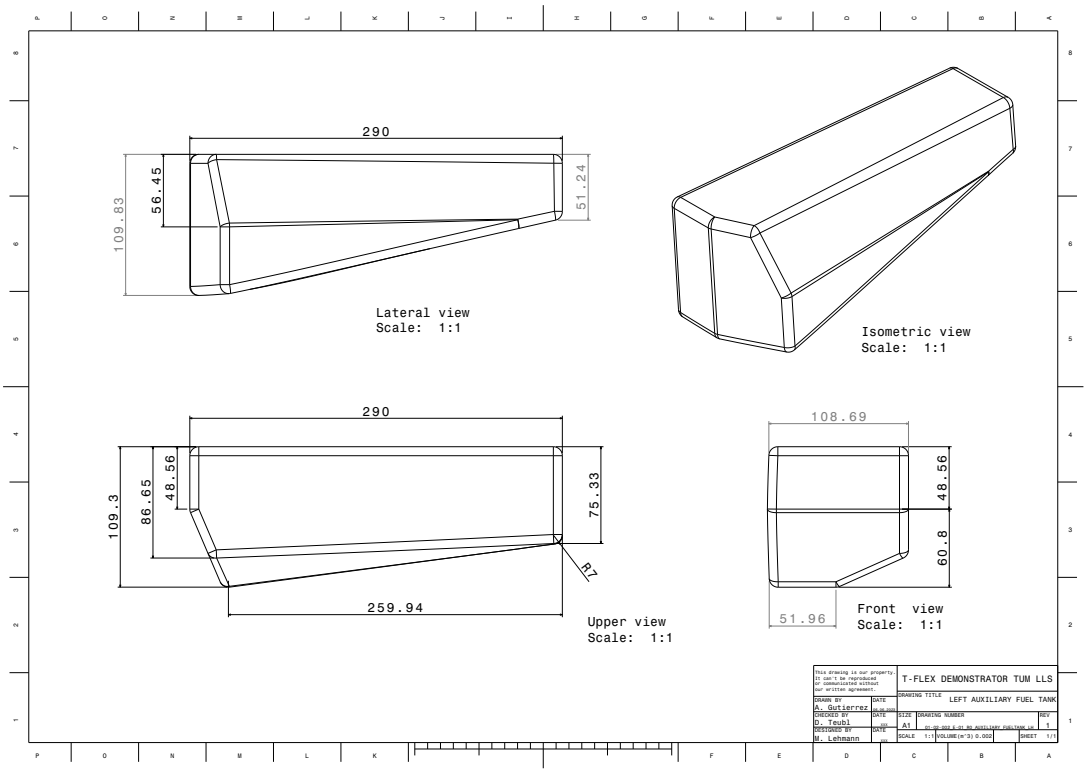


Figure 207: Technical drawing of one of the auxiliary Fuel Tank

Fueling and refueling is made in a way, that auxiliary and the main tanks can be fueled or defueled independently. An additional fuel-pump and on-off control logic is created, to make fuel transfer possible in flight between the auxiliary and the main tanks. The refueling system is controlled via a dedicated PWM channel from the backup transmitter.

Thrust measurement system As one of the goals of the project is to use active control for drag reduction, drag measurement in-flight would be necessary. This requires the thrust created by the

engine to be measured. Conversely, the thrust is a difficult parameter to measure.

Several methods have been developed and tested, whereas the most reliable require multiple sensors throughout the engine. These provide values that can be used to calculate the corresponding thrust. Such are known as gas-generator methods and are suitable for large aircraft, which have been designed with extensive sensory network within the engine. In contrast, smaller propulsion units such as the B300F that powers the FLEXOP demonstrator aircraft do not have provisions for the installation of pressure and temperature probes. Thus, the required modifications to the engine's structure hinder the viability of gas-generator methods. As an alternative, simplified and swinging probe methods were considered. These require no sensors inside the engine but are limited to gross thrust measurement, not sufficient for drag determination. Further, brochure and acoustic-based methods were studied but due to limited data provided by the engine's manufacturer, calibration would be complex and limited reliability would be achievable. Consequently, the trunnion thrust method was the option chosen as the most viable for a thrust measurement system to be installed on the FLEXOP aircraft. This technique is usually not considered feasible for larger aircraft due to the high complexity of the engine attachments, including multiple connection points as well as cables, pipes and hoses that make load path determination difficult. However, for a small aircraft, the attachment structure can be significantly simplified without affecting other systems and having higher design flexibility. As a result, the trunnion thrust method is suitable for this type of aircraft. Accordingly, a new attachment structure between the B300F engine and the aircraft's body was designed, whereas particular attention was given to obtaining a well-defined path for load transmission. More specifically, the structure was designed to form a statically determinate system when modeled in the aircraft's XZ-plane (symmetry plane). Thus, it is possible to determine the thrust force by measuring the load at a single support point with a load cell. Moreover, alternatives that kept the measuring system simple were preferred. For this reason, interference from varying vertical and lateral force components during flight maneuvers is not counteracted by implementing multi-axial load cells or devices that offer compensation for off-center and lateral loading. Instead, the support to which the load cell is installed was designed to only transmit forces in the measurement direction. This was achieved by implementing the support as two heim joints with the s-beam load cell installed between them. Thus, sensory complexity was kept low and bulky and heavy electronic components were avoided.

Additionally, the mentioned support was placed in the aircraft's symmetry plane to limit effects caused by thermal expansion, which have been a factor in previous attempts to implement the trunnion thrust method. In comparison, the other support points were implemented as rolling-element bearings placed on the sides of the structure. This allowed increased lateral stiffness for safer handling of the unit during maintenance operation but maintained the mechanical characteristics in the symmetry plane. Also, the configuration allows for low friction, which has been identified in previous projects as crucial for limiting the bending moments transmitted by the bearings and for allowing precise measurements. The final design is displayed by figure Figure 208. The structure was also designed with high measurement accuracy as a goal. For this reason, the positioning of the components was defined such that it minimizes errors. In fact, due to the ratio between the relative distances of the support points to the engine's center, all errors induced by the load cell and the analog-to-digital (ADC) conversion are nearly halved. This was shown by an error estimation performed using data provided by the load cell's manufacturer to predict the deviation between the actual applied thrust and the expected measured value, as shown in Figure 209. According to the diagram, the system would deliver an accuracy of about 0.4N and better at lower thrust levels. However, this estimate accounts only for load cell errors as well as ADC quantization and amplifier drift error. Therefore, additional influences such as higher temperature oscillations, manufacturing tolerances and further errors from the ADC (e.g. noise) may lead to lower performance and must be considered for a more accurate and extended prediction.

At the moment of writing, the system has already been compared to a measurement with a running engine on a static thrust measurement stand. The comparison graph can be found in Figure 210. The maximum deviation at full thrust was found to be around 5N, or 1.7% (the spike in the graph is due

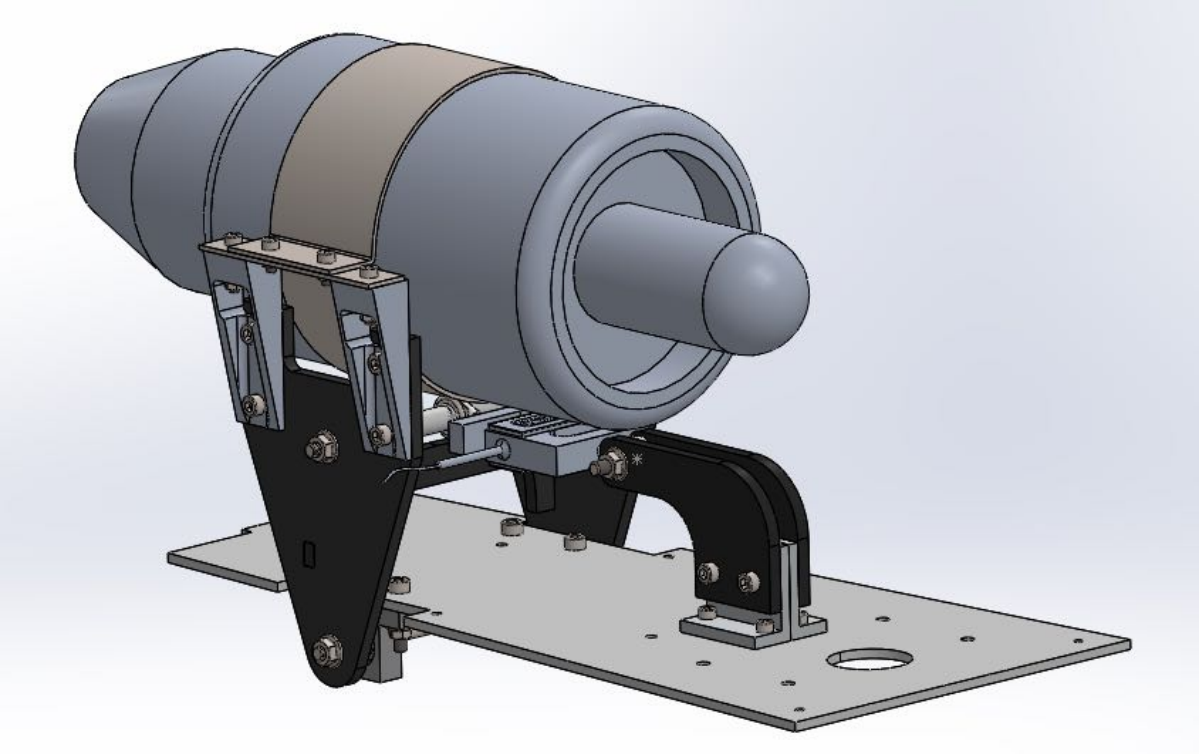


Figure 208: Thrust measurement unit with the mounted engine

to misalignment of the systems). This, however, does not take any possible misalignment of the two systems or the deflection of the engine stand. These sources of errors are currently under investigation and the accuracy is expected to be improved further on.

The thrust measurement system is expected to be flight tested within the next three months.

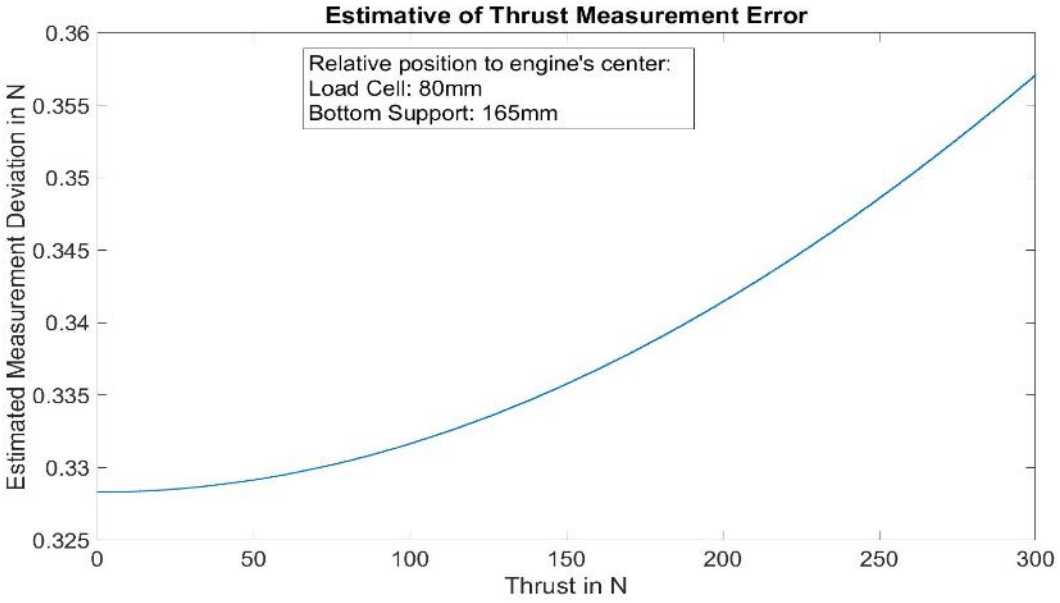


Figure 209: Estimated measurement error at chosen component positions for entire thrust range

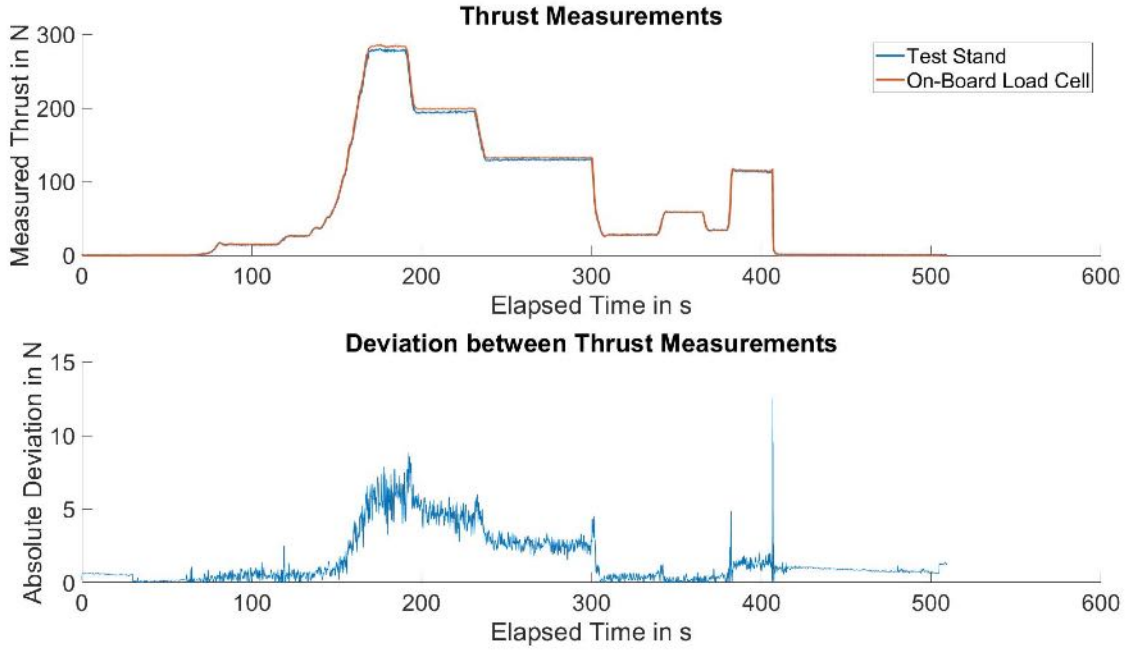


Figure 210: Comparison of on-board and static thrust measurements

Flutter stopper mechanism Another activity that relates to preparing the demonstrator for flight test campaign is the development of an emergency solution for the upcoming flutter tests. The idea was to be able to reduce risk of losing the demonstrator by creating a device that would completely change the resonant frequency of the flutter wing (-1), stopping it for flutter. The pilot would trigger this device in case of extreme wing flutter. Currently, the device is being designed and tests are being planned for

the upcoming winter.

Remote Control System As one of the cause for the loss of aircraft with T-FLEX, the decision were made to put more effort for testing and upgrade of our remote control system choice for the P-FLEX. The initial testing results are mentioned in [123] in the ”Remote Control System range test” section.

The decision is made to use JETI system for both main and backup RC system, but use a DS-24 transmitter as main, and use a DC-16 with NG capabilities for the backup.

For normal 2.4Ghz antennas, *REX3* modules were chosen with additional ”balloon” antennas for increased range and quality. The backup system got a *DUPLEX Rsat 900 NG* receiver as well.

The main reason to use JETI system as both main and backup system is the similarity in programming and configuration capability. Along with that, JETI support Lua scripting capabilities, which allows us to make custom functionalities on top of the base capabilities of the system, which none of the other systems which we tested are offering at the time.

Along with the increased range and stability a secondary goal were with the RC system update, to allow enough feedback directly to the pilots. They should be able to fully control the aircraft as a normal RC plane, in case there is a major connection loss between either on the communication radio, or between the aircraft and the ground control station. For that, additional sensors were introduced, and redundancy were kept for the main and backup system.

Each system got direct airspeed sensors, variometer sensor for altitude feedback. On top of that, the backup received a GPS sensors, and fuel-flow sensors, to be able to monitor and control the fuel transfer functionality.

Fail-Safe System During the investigation of the accident, a major cause for the full-loss of aircraft was the unintentional fire.To mitigate that, 2 additional changes were made on the overall system.An additional hardware block were introduced to enable/disable ignition power for the engine, and a software change got introduced to disable engine start command after certain conditions.The original chute-release mechanism were kept the similar.

Chute release system The chute release mechanism has two roles.First, release the chute anytime, the right conditions meet.Second, shut down the running engine.The action needs to happen at the same time, to prevent the melting of the parachute itself from the engine exhaust.

For the mechanism itself, 2 opto-switches - JETI SP06, and electric battery switch - EMCOTEC DualBat DPSI Micro - is used.The opto-switches are cutting the power line on the low-side (ground), based on the input signal from the respective transmitter system.The electro-magnet of the parachute is directly connected to the output of the batter switch.The output ground of the battery switch is connected to a safety line of the ECU, which will shut down an operational engine, if the safety line is not on same voltage as the ECU ground.

The system will terminate or release the parachute if:

- Both 2s battery voltage drops too low — battery loss
- Both RC system sends a terminate command
- Both RC system loose reception
- The system stays in terminated/release state after it is released

Because the opto-switches are cutting the ground, the voltage of the safety line will deviate from the ECU ground when the system is terminated.

Ignition switch The Ignition mechanism uses similar layout as the Chute release, meaning it has one opto switch controlled by each RC subsystem, and a two-input power switch, which can be activated via a magnet.Since the engine requires a dedicated 12V input for ignition, this power-line is enabled/disabled via the Ignition switch.

Similar opto switches are used as with the chute release, but ZEPSUS dual input magnetic switch were used, since that can handle 12V and high current flows.

In practice, this should behave similarly to the chute release mechanism, expect for staying deactivated, since this module has a built-in memory functionality.Based on manufacturing data, the memory functionality work for some seconds.

Improvements in power system (TUM) Past flight tests showed the need for the ability to put the FLEXOP demonstrator into a power-saving stand-by mode that allows extended waiting times with quick reaction times to use unexpectedly opening flight windows. The past efforts addressed these two issues.

In order to implement a stand-by mode for the aircraft, the power consumption of the demonstrator needed to be reduced to a practical amount by selectively shutting down components that feature a high power consumption, should not be operated idle for extended times and/or have a quick and uncomplicated boot-up process. The power supply system has to be capable to either run the remaining components over an extended period of time or capable of keeping the components running while batteries are being swapped. After an analysis of the existing system and iterative review of different possibilities the following measures were decided upon:

1. Adding a circuit breaker in the power line between one 2S-batterie and the power-distribution board.
2. Rerouting the cable supplying the RX-MUX-II-boards to the splitting point before the circuit breaker.

Thus, in order to put the demonstrator into a power-saving stand-by mode, the following main steps need to be performed:

1. Moving the power switches of the FBG-interrogators to ”OFF”-position.
2. Disconnecting the 3S battery.
3. Removing the circuit-breaker of the first 2S-battery.
4. Disconnecting the second 2S-battery.

In this state, the 6S-battery is only powering the Raspberry Pie of the FCC flight stack, which can be supplied for several hours. In order to start up the demonstrator for flight tests, the above steps are undone in reverse order.

After the implementation of the changes to the power system, a new landing gear was rigged in. The new landing gear setup features linear actuators for retraction and deployment, as well as drum brakes that were expected to have a higher holding force and less wear on the tyres than the stamp brakes used before. The breaks work with three different voltage levels, i.e. 12 V, 7.2 V as well as 6 V. The

different voltage levels are supplied by the 3S-battery and the 2S-batteries respectively. In order to provide a supply voltage of 6 V, a DC/DC-converter was introduced that supplied both gear system. During testing the brake servos did not operate reliably. Investigations on the system yielded a signal-cross-talk from one servo signal line to another. The problem was solved by introduction of another DC/DC-converter.

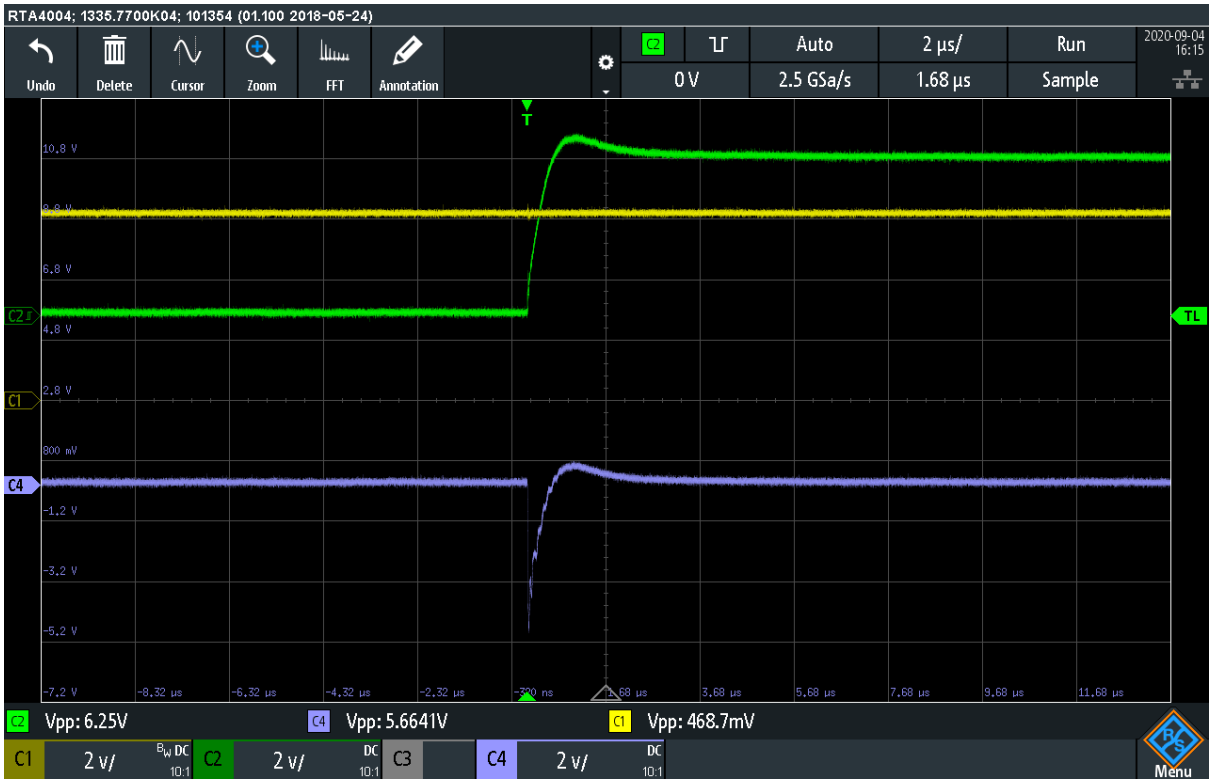


Figure 211: Close look on the cross-talk, when only one step-down converter was used for both actuator. The peak value of the noise overshoots the standard TTL logical threasholds

During the rebuilt, a similar layout was chosen to the power distribution board as we had before.

Main changes compared to the old system:

- The plates are $6mm$ aluminum, instead of $3-4mm$ copper
- a 5V board is introduced
- the two 2S board got separated to 4 board
- the distance between the individual board got doubled

The power inputs and their respective consumers are visible in table 26. With the splitting of the 2S boards, it made possible to have the main system operation as soon as either battery is plugged into the system, but not the power consumers.

The Direct Drive system is powered directly via dedicated cables from a 12S battery pack, which two 6S in parallel connection. The 'Flutter-stopper' servo motor power supply is connected via power line of the CAN-bus for the Direct Drive.

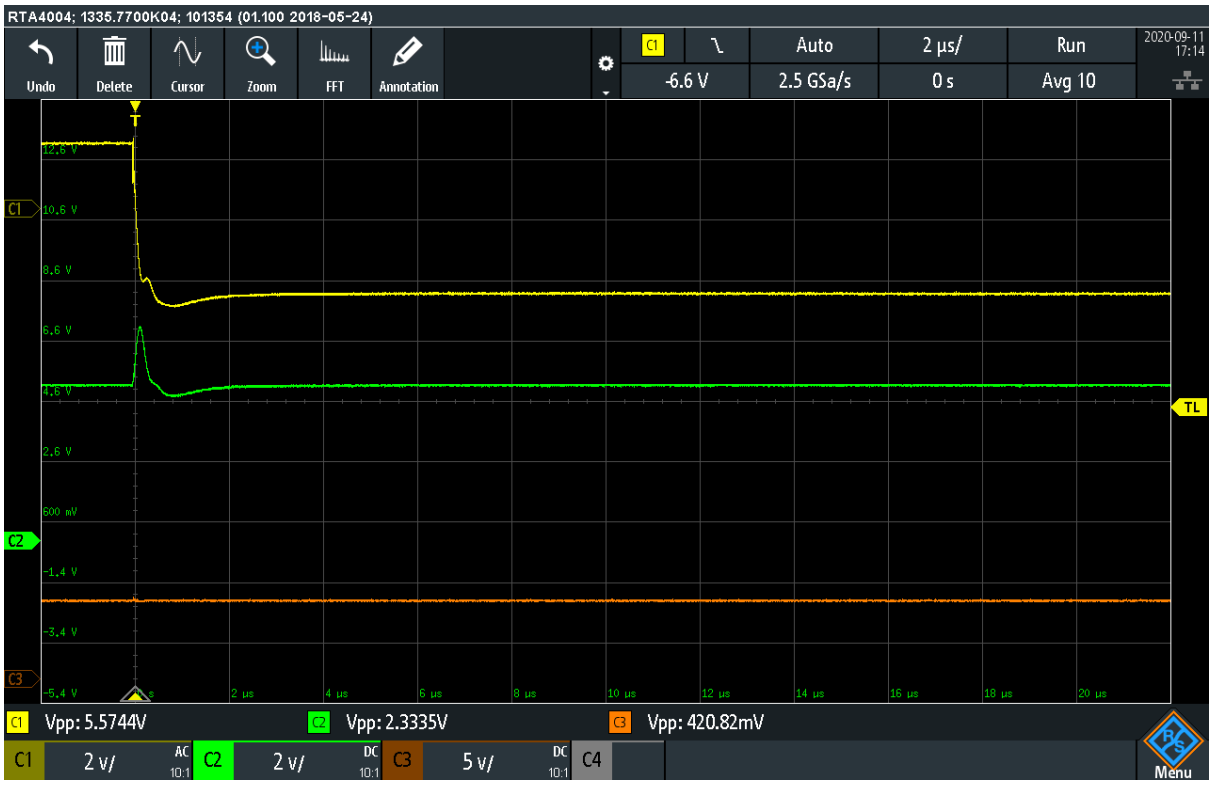


Figure 212: Crosstalk still visible with dedicated step-down converters. The noise peak is still high, but it does not show visible error on the system itself

6S — 24V	3S —12V	2S — 8.2V #1	P1	2S — 8.2V #2	P2	5V
FCC	Ignition	RX-MUX-II #1	LF_1	RX-MUX-II #1	LF_2	OBC-II
5V	Fuel pump	RX-MUX-II #2	RF_1	RX-MUX-II #2	RF_2	Telem_1
		RC #1	LF_3	RC #1	LT_2	Telem_2
		RC #2	RF_3	RC #2	RT_2	
		chute	LT_1	chute	brake	
		ECU	LT_2		LFS	
			Tailwheel			
			RFS			

Table 26: Layout of the different power consumers and their connections,

Takeaways from the layout and designs:

- Splitting the 2S board made operation easier, and made possible to have longer wait time before takeoff, without discharging the main flight control batteries
- Using the regulated $5V$ input for the telemetry modules directly, made the wiring a bit more complex, but allowed the free power input to the telemetry modules. Previously they were powered directly from the FlightHAT module, where the internal $5V$ regulator had way more limited power output.
- Adjusting the spacing between the aluminium plates, made it possible to have a nicer way of wiring than earlier 215

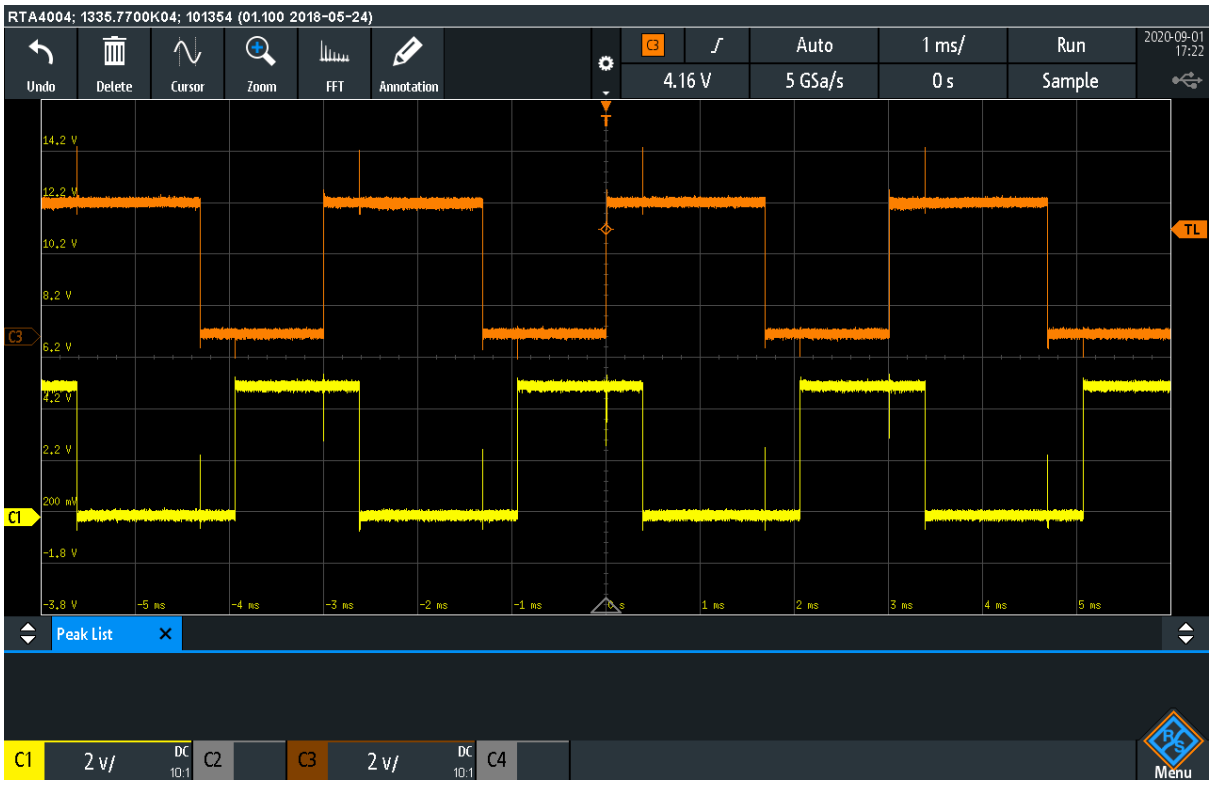


Figure 213: Visible cross talk on the independently generated PWM lines, when actuators are attached

- Having the ECU and the chute directly connected to the battery plates, made possible to confirm ECU communication and chute release checks, without running the overall system.

major design drive for the avionics plate was, to be able to easily remove it from the fuselage. For that, each cable and subsystem connection needed to be cut and ended in one or many physical, easy to access connectors.

late includes the following subsystems:

- FCC Stack
- Battery holder
- Power distribution board
- RC system
- Fail-Safe system

With the RC system and most of their sub-components directly installed on the avionics plate, the number of connector were minimized. Both part of the Fail-Safe system is connected to the avionics plate via one or more secure connection point. With that, replacement or easy repair of the individual modules are possible.

To keep modularity, the connectors between the avionics plate and the rest of the system were designed based around different subsystems, and their respective location. The 6 main connection point are the following:

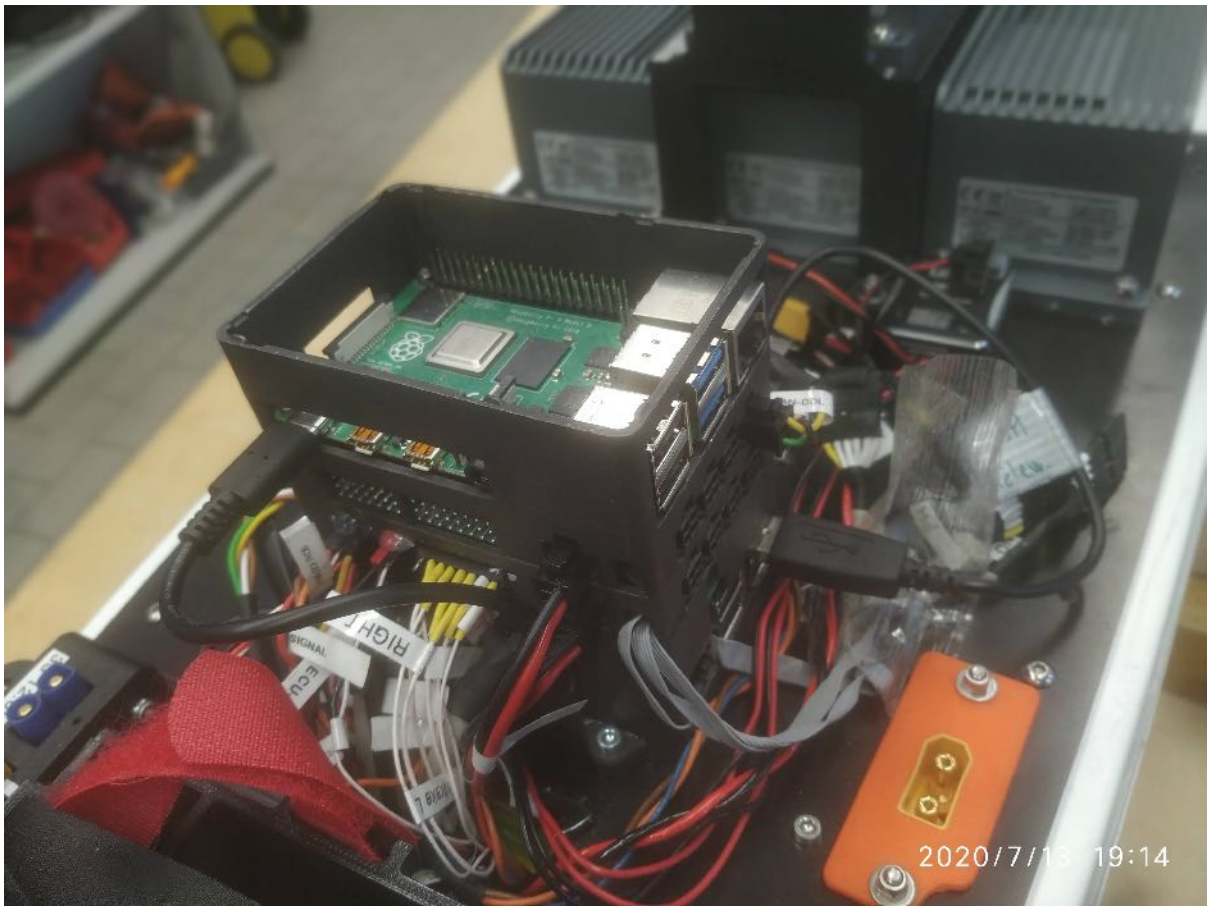


Figure 214: Secondary on-board computer on top of the existing flight-stack

- Propulsion system - D-sub
- Sensors and telemetry - D-sub
- Left tail - D-sub
- Right tail - D-sub
- Left wing
- Right wing

For the 4 main connector, replaceable pin D-sub connectors were used. Although, a new tool and technique is needed to be learned for setting the connectors, it turned to be faster and easier to set up all the connections. Moreover, it was possible and easy to fix single wires during integration if it was needed.

The left and right wing connections contains the actuator power and signal cables, CAN cable for the IMU's, CAN and power cable for the direct drive system.

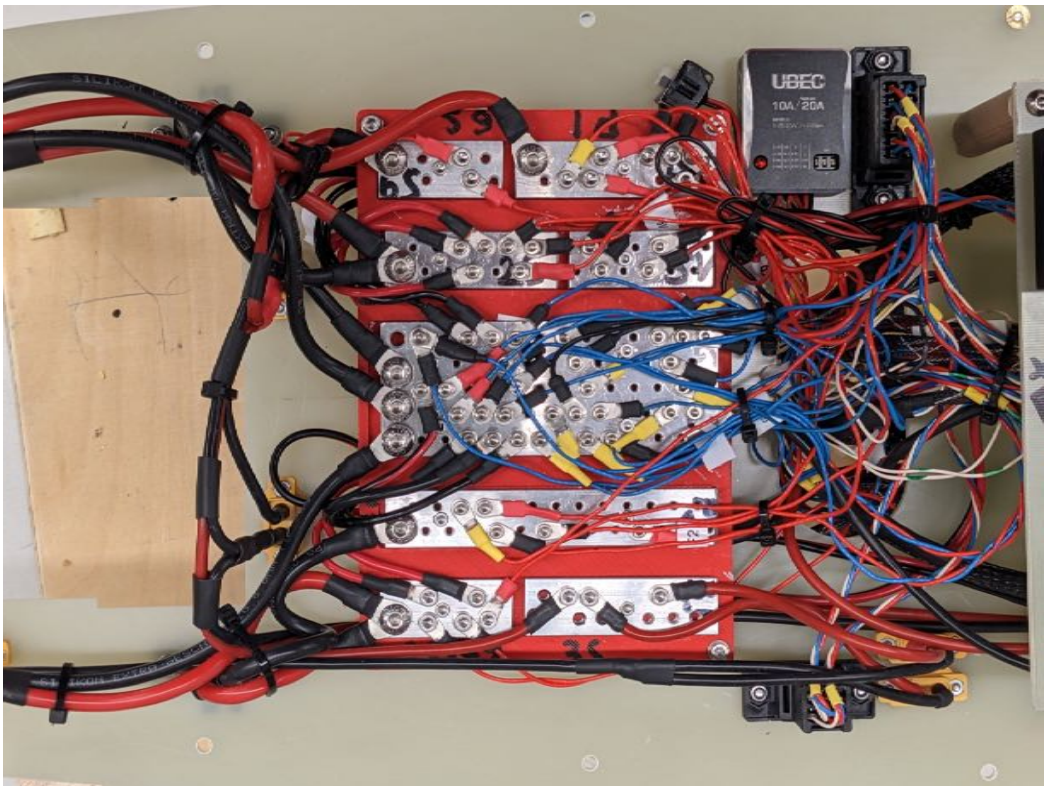
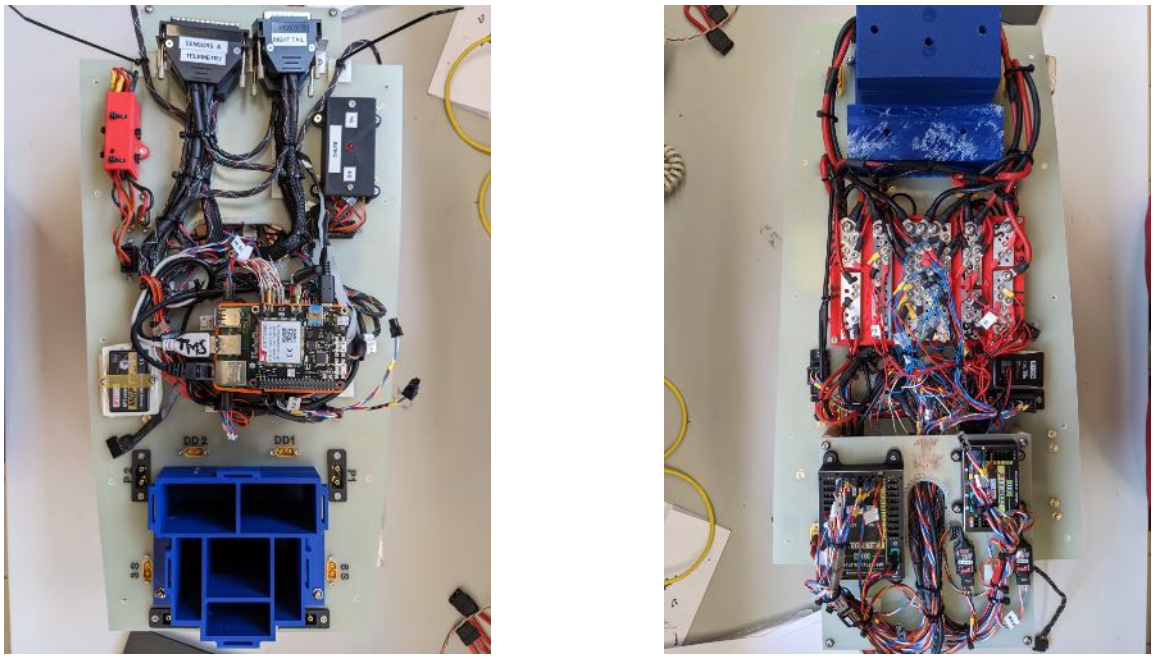


Figure 215: Look of the already integrated design

Avionics improvements (SZTAKI) The demonstrator used in the FLEXOP project was constructed with a custom Flight Control Computer (FCC) developed by SZTAKI. During the previous project, we encountered the limitations of the FCC's capabilities, necessitating the need for improvements. The most significant improvement was demanded by the safety-critical module, called RX-MUX-II. We required additional input and output channel handling, a broader range of interfaces, and enhanced computational capabilities. Hence, the complete module redesign became imperative. Following the testing of the initial manufactured components, the global chip shortage posed a significant obstacle, preventing further production. Consequently, we had to undertake a comprehensive hardware redesign, leveraging the limited pool of available components in the market, all while ensuring compatibility with the pre-existing embedded software. figure 217 shows the top side, figure 218 the bottom side of the second redesign of the RX-MUX-II module.

During the project we undertook a redesign of the aircraft's Remote Control (RC) structure. This involved replacing the previous PPM communication with the digital EX Bus protocol for communication between the RC receiver and the RX-MUX-II. As part of this process, we made the decision to exclude the Sbus implementation, opting for a Jeti system for both primary and backup functions. The revised RC system provided increased channel capacity and accommodated modifications in the aircraft's configuration, leading to an overhaul of the entire RC structure. These changes subsequently triggered adjustments in other software components as well.

The new RC system offers 8 additional channels for a total of 24. Most of the new channels were used to split up functions on the aircraft that were previously combined to one, and the complete channel order was re-assigned, to make it more transparent. The RX-MUX-II was updated to handle the additional channels. Since now we had the ability, we upgraded the RC system selection method. The now imple-



(a) Avionics plate top view

(b) Avionics plate bottom view

Figure 216: Fully integrated avionics plate before GVT

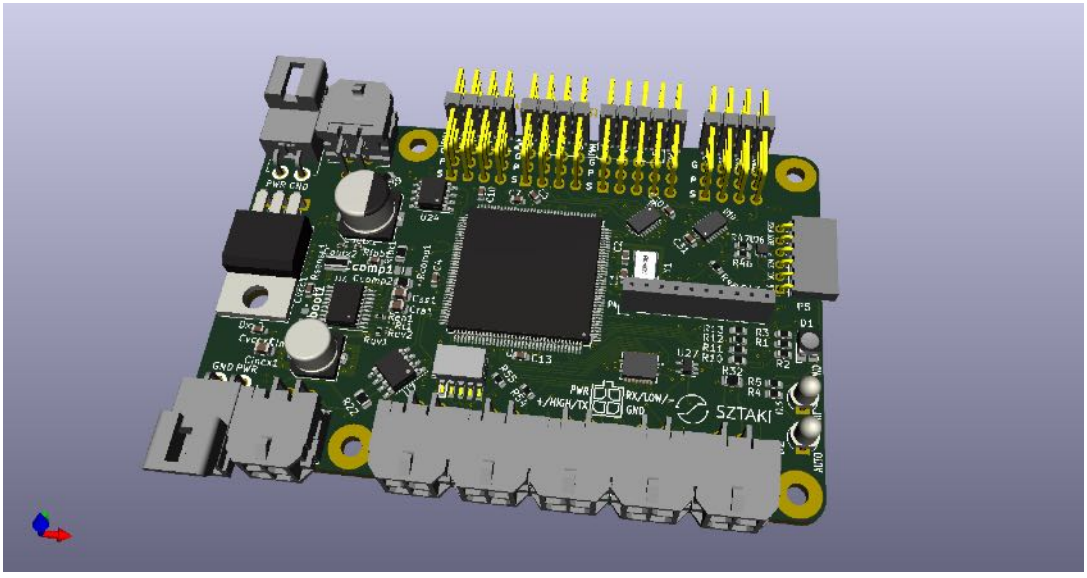


Figure 217: RX-MUX-II v2.2 top

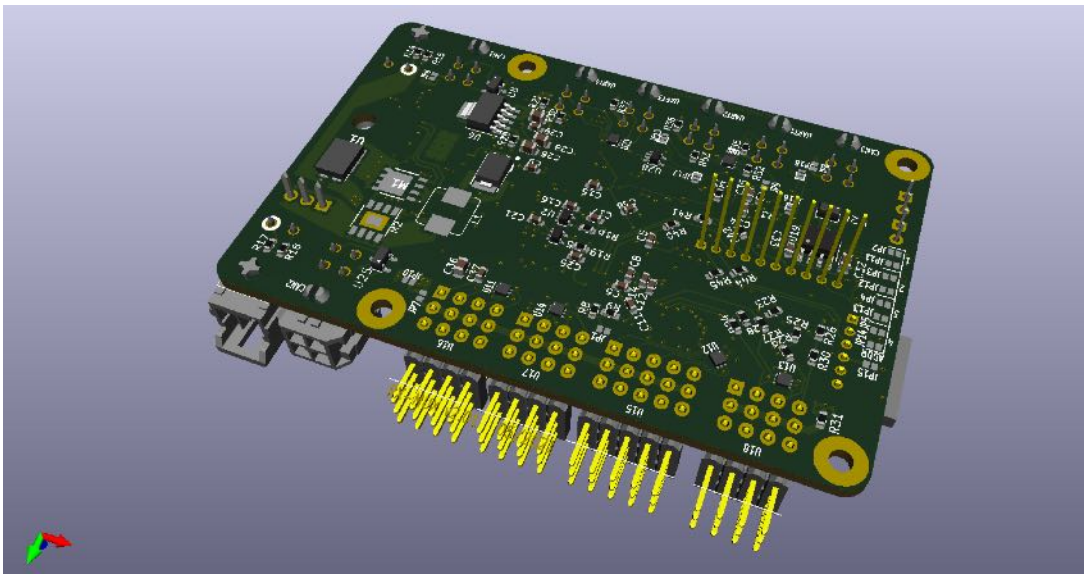


Figure 218: RX-MUX-II v2.2 bottom

mentation allows the pilot monitoring to take control of the aircraft without the other pilot's cooperation. However, the pilot currently controlling the aircraft, cannot give the control to the pilot monitoring. If this is attempted, control is only transferred after the pilot monitoring also flips the correspondig switch on his transmitter.

To control the planned CAN servos in the -3 wing, the software component responsible for handling them was heavily tested, finalized and added to the code. After the plans changed regarding the advanced wing, the CAN servo handling feature had to be slightly modified. This was needed so that we can use a direct drive and a CAN servo, on the same CAN bus.

The direct drive also required additional software changes. To communicate with the drive controller, a new software module was added. This new module consists of a CANopen stack, specifically CANopenNode. CANopen is a high level communication protocol and device profile specification, that is usually used in automation. It uses an ordinary CAN bus for the physical and data link layers. This modification turns the RX-MUX-II into a CANopen capable device. The CANopen network in our case only consist of 1 RX-MUX-II and 1 Direct Drive. A simple custom device description was made for the RX-MUX-II, just enough to be able to control the necessary functions of the drive. RX-MUX-II are responsible for one direct drive each. One of the three available CAN ports on the RX-MUX-II was modified to use CANopen.

On top of this, a state machine was implemented, that establishes communication with the drive, configures the required parameters, and tries to turn on the motor as soon as possible. If the drive detects a problem and stops, the RX-MUX-II checks the error code, and if the error was declared safe (for example momentary over current), it will attempt a restart. All other maskable fault reactions were disabled in the direct drive, to avoid any unnecessary shutdowns.

The architecture of the RX-MUX-II software illustrated in Figure 219.

In the previous configuration, the autopilot module consisted of a Raspberry Pi 3B+ running an older version of the Raspbian OS that had been patched for real-time capabilities. Following the implementation of real-time kernel modifications to the new Raspberry Pi OS and numerous testing iterations, we proceeded to upgrade from the Raspberry Pi 3B+ to a Raspberry Pi 4 for the autopilot module. Addi-

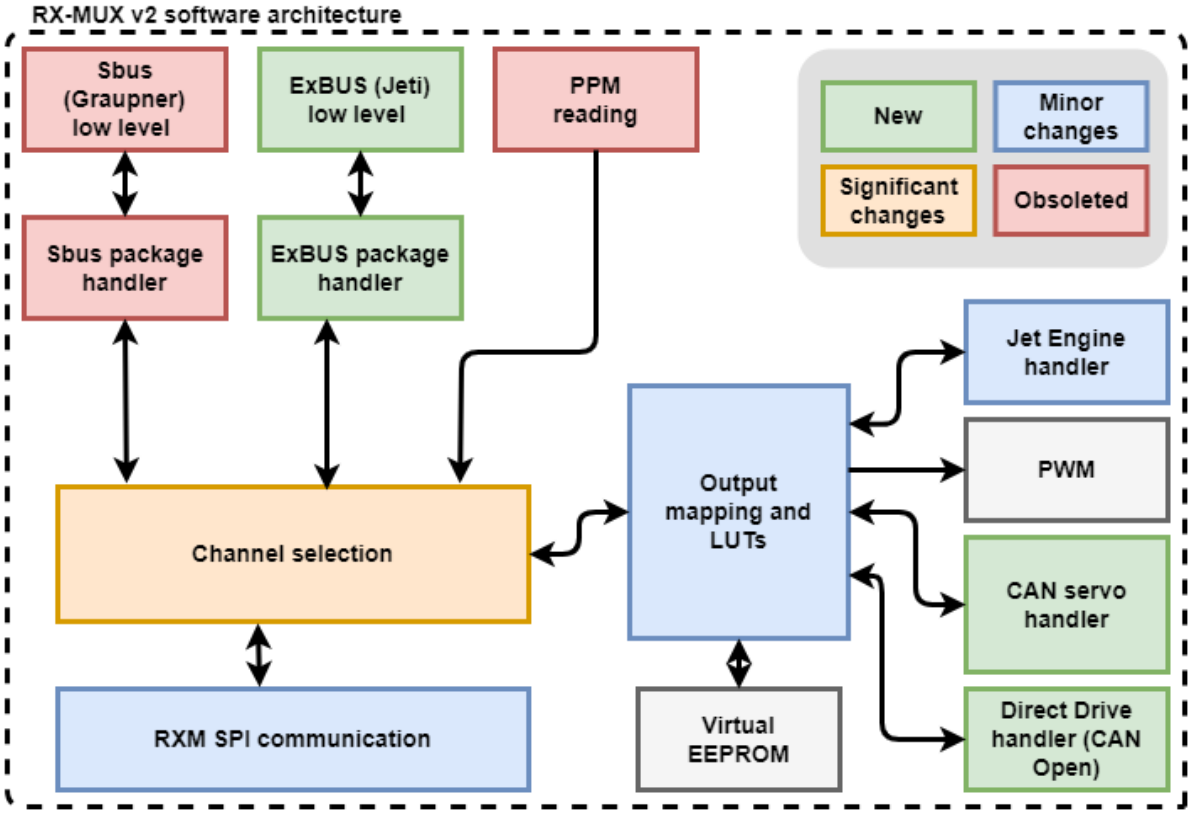


Figure 219: RX-MUX-II v2.2 software architecture

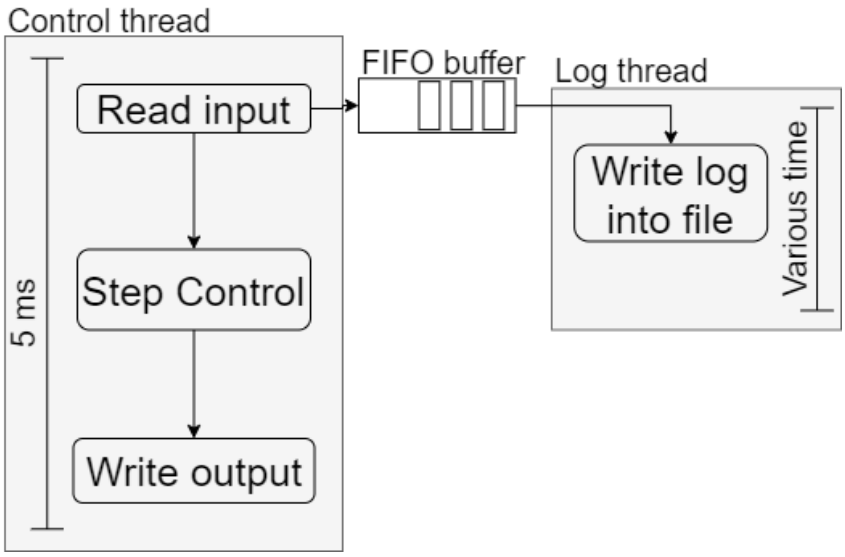


Figure 220: RX-MUX-II v2.2 software architecture

tionally, we made alterations to the logging approach of the autopilot module. This adjustment became necessary as the operating system struggled to write data to the SD card in real-time. To address this issue, we introduced a FiFo buffer that retained the data within the system's Random Access Memory (RAM) while the SD card was occupied. Subsequently, a dedicated thread, operating on a separate timing cycle, would write the accumulated log data to the SD card once access to the resource was granted. The logging concept shown in figure 220.

We further enhanced the autopilot's functionality by enabling parameter adjustments through the telemetry software. This modification aimed to simplify flight tests, eliminating the need for manual software alterations during flight. Previously, adjustments required landing and changing the software to incorporate different parameters. Further details about the autopilot interface in the telemetry section.

Improvements in telemetry system (SZTAKI and TUM) To streamline flight tests, we have developed a user-friendly graphical interface for the autopilot. This interface empowers ground test engineers with the capability to modify autopilot parameters during flight. We integrated it into the Mission Planner software, which serves as the primary telemetry system guiding the aircraft for the flight team. We made the baseline functions selectable, enabling us to test each one individually and modify the autopilot behavior without needing to land. We added buttons for selecting injectable signals, facilitating system identification tests. We also incorporated buttons for altitude and velocity selection, which was highly valuable during flutter damping tests as well. The user interface in Mission Planner shown in figure 221 and the interface for baseline and flutter suppression on figure 222.

Former experience has shown, that the focus of interest in different data sources shifts between flight tests. While initial flight tests focus on system checks, data yielding information about the system performance such as temperatures, currents, voltages and fuel flows are of greatest interest to assure safe flying. With increasing routine and experience with the demonstrator, focus gradually shifts to different data such as airspeed or altitude to attain and keep the planned test conditions or the identification of different modes. This shift in focus also manifests itself by the adaption of the data displays and mode of visualization. The current display of the Engineering Data Link developed in Mathworks Matlab does neither offer the necessary flexibility to change display layouts fast nor does it offer a great variety of different modes of display. In order to improve the flight test efficiency by usage of more flexible dis-

Figure 221: Autopilot graphical interface in Mission Planner

plays, NASA's OpenMCT framework was implemented and adapted for flight testing of the FLEXOP flight demonstrator. Expected advantages of the new visualization framework are flexible adjustment of data displays, saving of different views that can be switched easily, a wide variety of widgets already available, data "playback" functionality that greatly improves and facilitates flight test debriefings as well as the increase of flight test participants by providing flight data live to remote participants, that can provide additional expertise.

A working state of August 20 is displayed in the Figure. The moving graphs are widgets that can be adjusted in size and colour as well as types of data displayed.

To date the functionality that has been tested with the FLEXOP flight demonstrator is the display of different modes and normalized eigenfrequencies identified by the secondary Raspberry Pi developed by DLR Göttingen using OpenMCT. Flight monitoring functionality – including safety critical - and flight test adjustment ability has been field-tested and validated using the DG-800 S flying testbed of LLS. This type has great resemblance with the FLEXOP configuration (sailplane with dorsal turbine) and has already been employed for pilot training. Combined with an antenna-tracker, which provides a high-bandwidth connection to the testbed using 5GHz-Wifi, data collected for sytem identification of rigid modes has been streamed down and displayed live in OpenMCT. During flight tests conducted, the reliability and flexibility of the framework was proven: E.g. the data visualization enabled the identification of a sensor failure, which allowed the adjustment of the fight test routine including a change of data displayed. Furthermore, the display proved to be so realiable, that a reduction of safety margins concerning fuel available was possible, which resulted in a near-optimal use of flight time. Initial tests showed the possibility to increase the number of flight test participants by streaming available data to a server, from which it is accessible remotely.

Upcoming efforts will target the implementation of a 5GHz Wifi, high-bandwidth downlink from the FLEXOP flight demonstrator and development of display templates required for future flight tests. Therefore, a Wifi-connection will be established to the secondary RaspberryPie, which will send down data necessary to duplicate the Engineering Data Link. The amount of data will gradually be increased

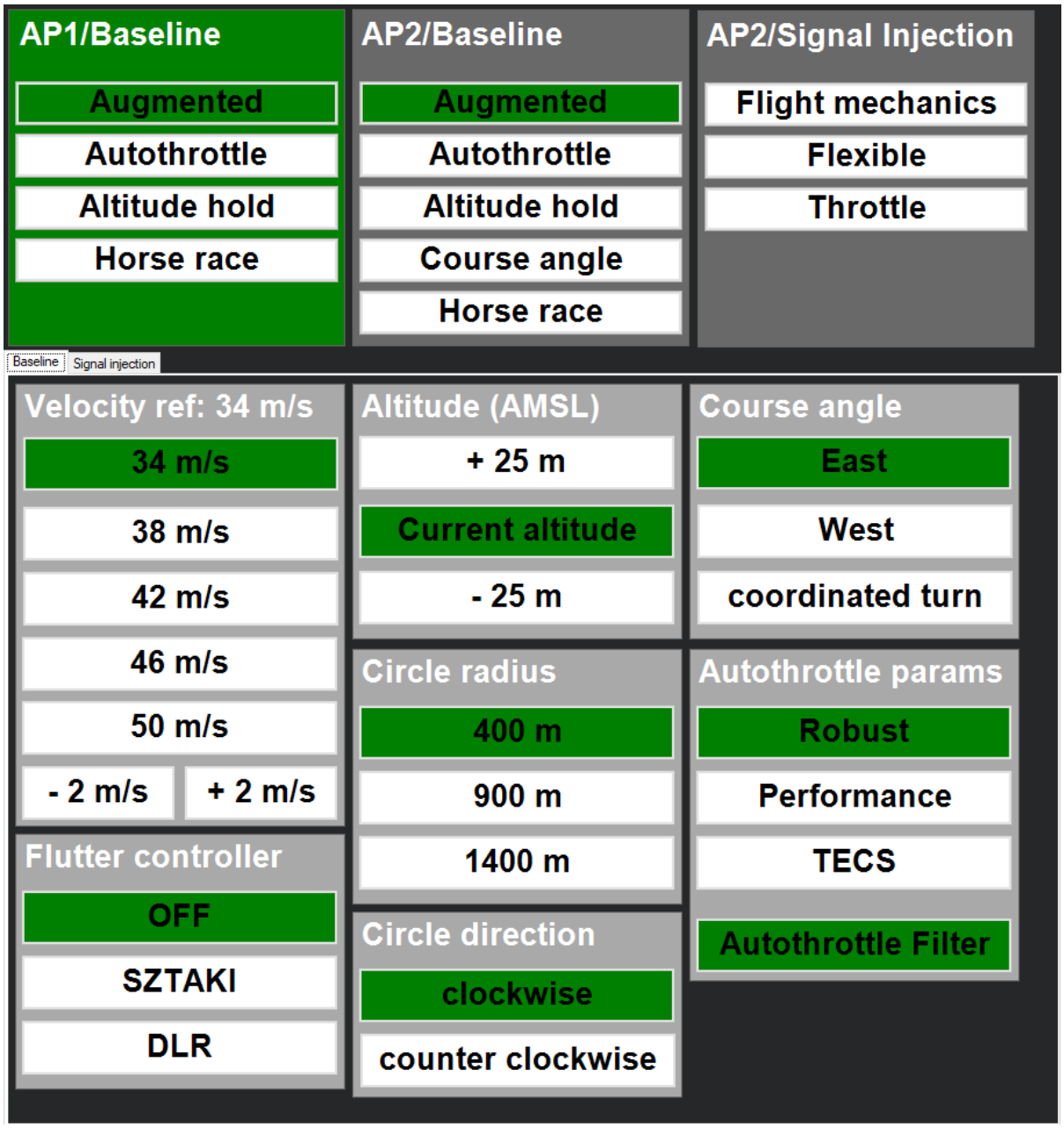


Figure 222: Autopilot interface for baseline testing and flutter suppression tests

Figure 223: Display of NASA's OpenMCT visualization framework during flight tests using the DG-800 S flying testbed

in order to provide flight test relevant data to the operators. On the hardware-side, integration of servers and an LTE-router are ongoing and will enter consolidation and commissioning phase next.

In order to make full use of the capabilities of OpenMCT, an antenna-tracker was field-tested and commissioned for flight tests using the same DG-800 S flying testbed as employed for testing OpenMCT. The antenna tracker depicted in the figure below.

The tracker uses its own as well as the UAV's position to align the antenna to a position facing the UAV in air. During flight tests, 400 values per second were received and processed without experiencing drop-outs. The testcases included distances of 700 m and more, as well as close inverted flybys to test the systems robustness in case a bad GPS-reception. Given the experience of related projects, it can be assumed that the system's capabilities are not maxed out yet.

Therefore, next efforts will focus on further testing of the system as well as increasing the traffic on the data link to use the antenna tracker to its full potential.

Prior to the crash, there were three telemetry module operational in the payload are, with similar physical setup. In each case a 3DR-Sikk 433Mhz telemetry module were used, with a half-wave dipole antenna. Two telemetry module were connected to the FlightHAT, and one additional to the OBC-II module. Figure 225b shows a similar module before integration into P-FLEX.

The RFD 868x 868Mhz module were chosen, as a plug-in replacement for the EDL telemetry. Due to a software reconfiguration, the EDL is moved to the OBC-II module. That made it also possible to reduce the number of telemetry modules by one. In theory, the increased bandwidth with the 868Mhz telemetry system, both data link can be streamed via the same module.

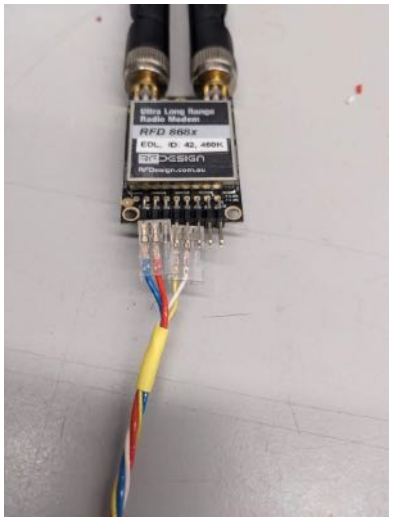
The chosen telemetry system 225a comes off-the-shelf with two attachable dipole antennas, which can cover the full 3D space around the aircraft, compared to a single dipole antenna, which will have good

Figure 224: The antenna tracker being prepared for DG-800 S flight tests

and bad reception block in the 3D space around the aircraft.

By using only two modules, in the small place in the payload area, the potential interference between each radio link is eliminated - in theory.

To help with the telemetry system reception range, the payload/avionics plate is made of glass-fiber instead of carbon-fiber, like in the previous iteration.



(a) 868Mhz telemetry antenna used in P-FLEX

(b) 433Mhz telemetry antenna used in T-FLEX, located next to the 868Mhz module in the payload area.

Figure 225: Telemetry units planned to be used

Air-data and IMU sensor mount updates During the flight test data analysis phase (with the flight test data from 2019) it was noticed that the angle of attack signal is corrupted with noise which is not visible in angle of sideslip. Figure 226 shows this difference in signal noise during flight path reconstruction of a flight segment. Angle of attack sensor appears to have a visible additional noise to it, which does not exist in the angle of sideslip. The sensor was checked in the wind tunnel and it was clear that the problem is not with the sensor itself as the spectral densities in both angles were the same (Figure 227, right). It was therefore postulated that maybe the mounting of the sensor is not rigid in the longitudinal plane.

The mounting of the pitot boom was therefore investigated. The mounting of sensor was done in a way that the air data boom would go through the nose section of the fuselage and then would be mounted on the payload rack board at the root (Figure 228 and Figure 229). It was then realized that the payload rack, which is a 3mm glass fibre board with many equipment mounted on it, would move vertically during manoeuvers and in this way would move the root mount of the boom as well. Considering that the middle point of the boom, which goes through the fuselage, acts as a rotation point, the actual sensor head therefore gets deflected (Figure 230). It was also recognised, that the main IMU sensor is also mounted on the flexible glass-fibre board. Therefore relocation of both main sensors (xSens and Aeroprobe) has to be done.

The air-data boom mount was upgraded by designing a new, rigid structure from carbon-fibre sandwich in the nose section of the fuselage (Figure 231). The purpose of the structure was to decouple the air-data boom mount from the rest of the payload rack and increase the stiffness of the point where the boom intersects the fuselage (the front wall). Solution was implemented.

In addition, the xSens was relocated onto a stiff mounting point next to the fuel tanks.

A single test flight has been done with the new sensor setup. The sensor error analysis of the new setup is not yet completed.

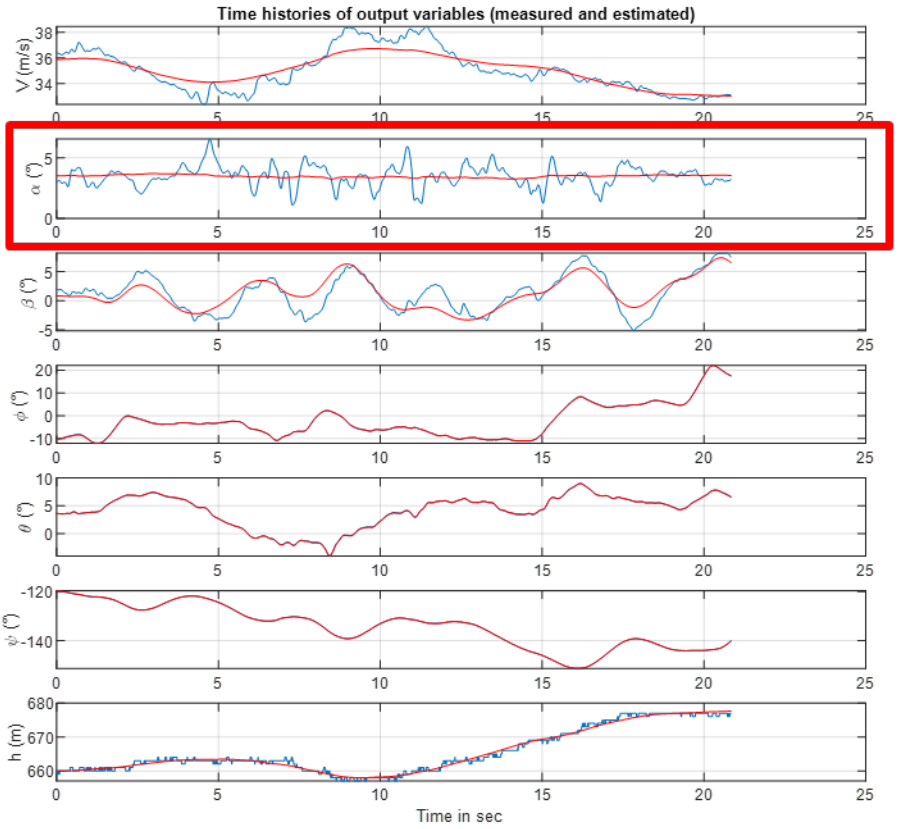


Figure 226: Data compatibility analysis (also known as flight path reconstruction). Ideally, the measured and estimated signals should match. Blue- measured signal, red- estimated signal. Clear difference in noise levels between angle of attack (alpha) and angle of sideslip (beta) can be seen

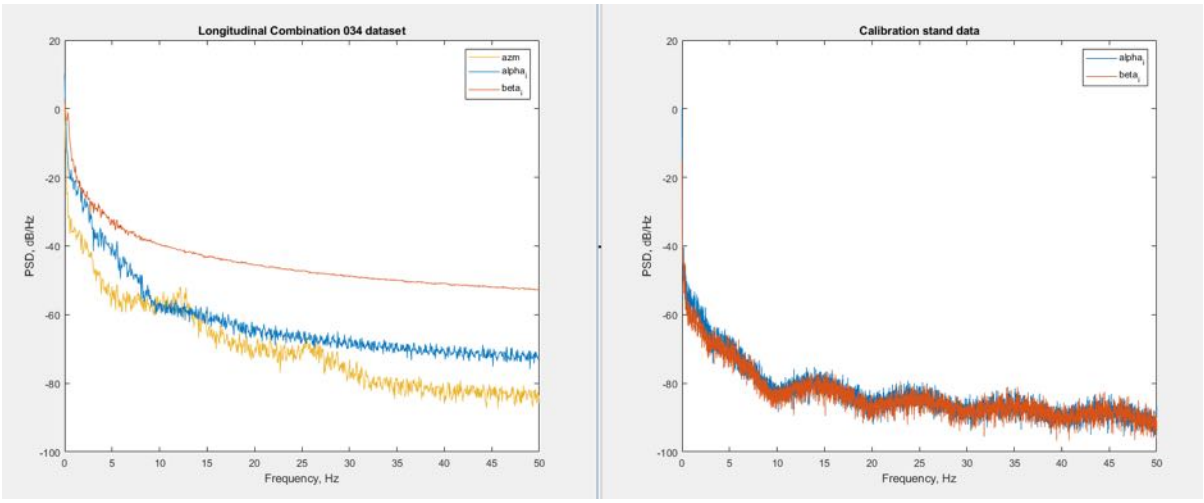


Figure 227: Angle of attack and angle of sideslip signal comparison from in-flight data (left) and wind-tunnel data (right)

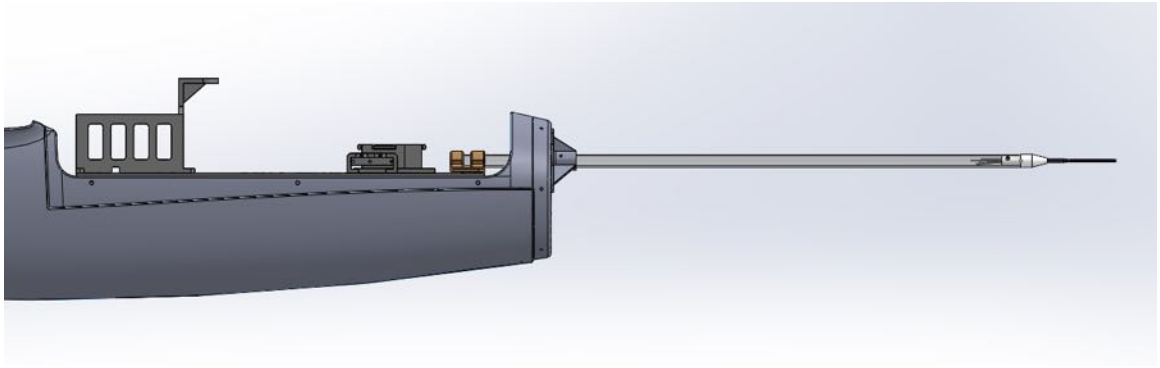


Figure 228: Air-data boom mount

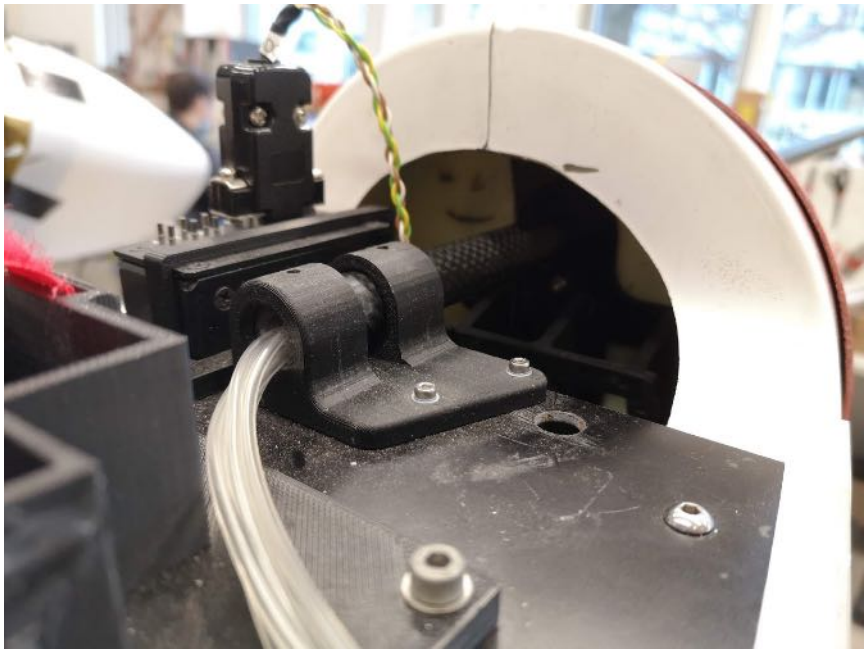


Figure 229: Air-data boom mount at the root

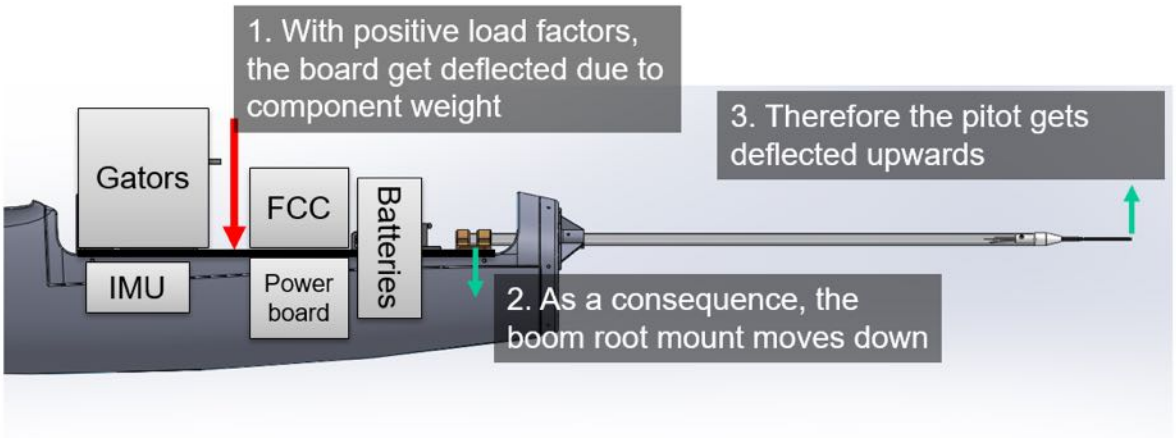


Figure 230: Air-data boom flexibility mechanism

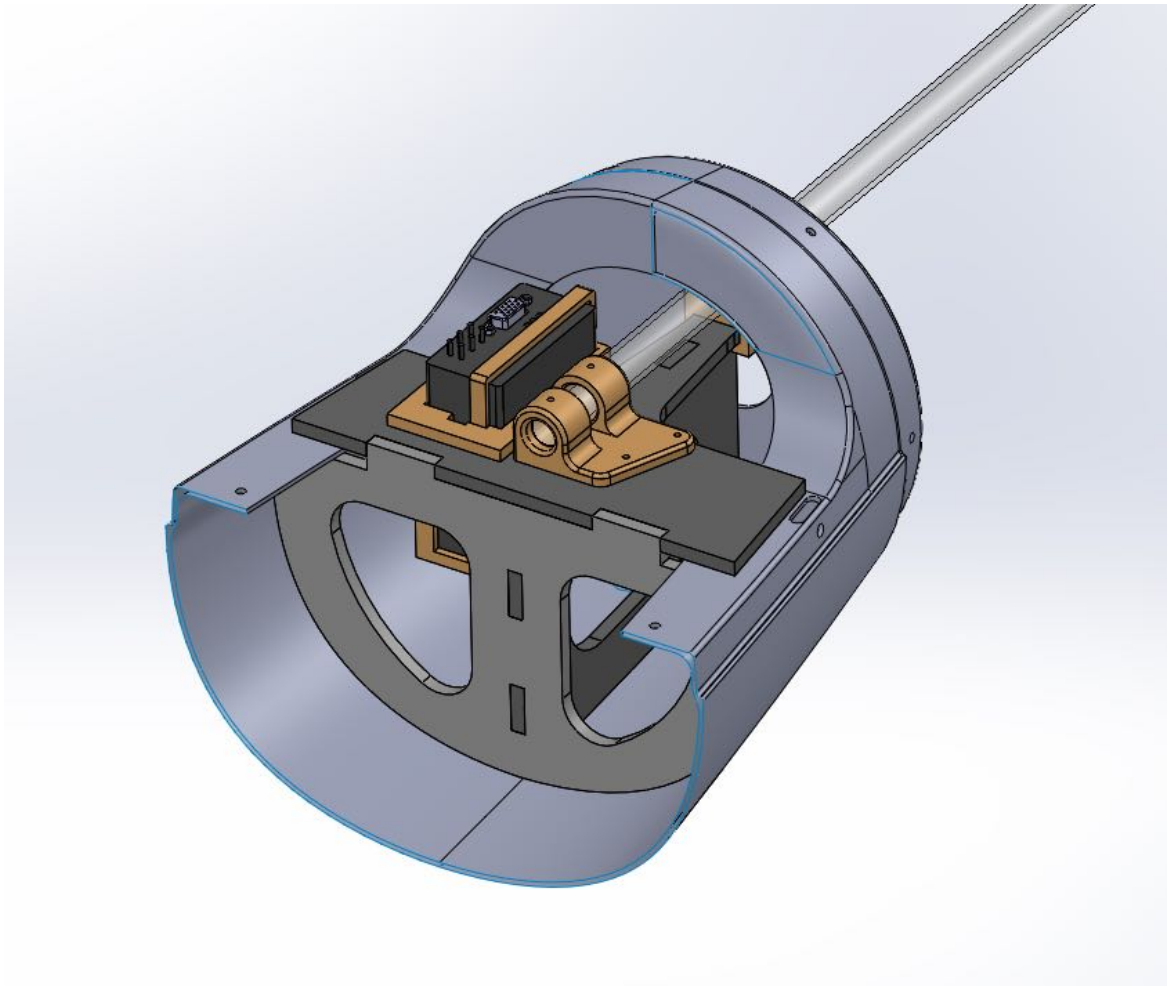


Figure 231: Upgraded air-data boom mount in the nose section of the fuselage

additional IMU sensors (SZTAKI) As an expansion for the existing build in IMU's in the wings, four additional IMU is installed. One in the front of the fuselage, one in the empennage, and one next to the V-tail spar to each side. Since the IMU's are using CAN protocol for communication, each of the additional IMU is just connected to either of the existing CAN network on the aircraft. Table 27 summary of the IMU location and CAN connection.

The additional IMU's will be used via the OMA during flight.

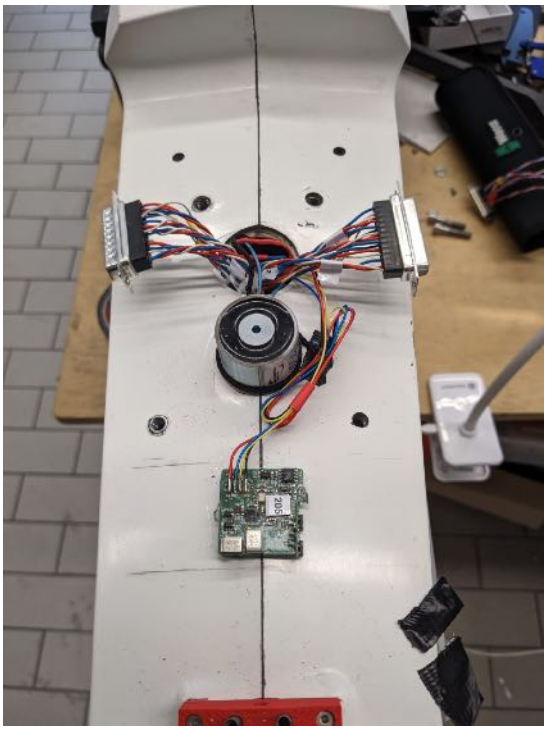


Figure 232: IMU located at the root of the V-Tails

ID	location	CAN bus connection	logID
184	left tail	left	13
191	fuselage nose	left	14
198	right tail	right	15
205	V-tail root	right	16

Table 27: Caption

Optical wing shape tracking Methodology to measure wing deformations in-flight was researched in a Master Thesis by Mr. Pablo Varillas Iglesias at TUM (guided by Julius Bartasevicius). The relevant subparagraphs are adjusted and included in this subparagraph.

The hardware used in the implementation of this wing shape measurement were the FLEXOP UAV T-FLEX and the two rear cameras. The two cameras are the central part of the hardware. The model is the Mobius HD Action Camera [77]. Its principal characteristics are the lightweight and small design. These features make the Mobius cameras a very good option for mounting them on a small UAV. The specifications of the Mobius camera are shown in table 28.

Table 28: Specifications of the Mobius HD Action Camera

Specification	**Value**
Size	61mm (L) x 34 mm (W) x 18 mm (H)
Weight	38 g
Max. resolution	1920x1080
Max. frame rate	60 fps
Battery	820 mAh
Autonomy	approx. 130 min
CMOS area	2304x1296
Resolution-frame config.	1920x1080p@30FPS, 1280x720p@60FPS

The flight test videos were recorded with the configuration 1920x1080p@30FPS. The camera saves the videos in the MOV video format with H.264/AVC1 video codec. Different modes such as time-lapse, photo, and video are available. Support for MicroSD Card up to 32 GB is supported [77].

The Mobius camera has only three control buttons to operate all functions. If more information is needed, refer to the instruction manual [76].

A CATIA 3D model of the T-FLEX was used to obtain the target's location in the ABFF. As mentioned before, the Mobius cameras are mounted on the tail of T-FLEX, aiming at the wings. The integration is achieved by a black 3D printed part.

Software

In this subparagraph, the development and implementation of the software are illustrated. The wing shape measurement software was programmed entirely in the Python programming language. Additional Matlab scripts were written to plot the obtained results because of Matlab's user-friendly plotting interface. The software was developed on a PC with the operative system (OS) Windows 10 Professional.

Git was used for the version control from the beginning of the development. The Python extension was installed for the support of the programming language. The Excel Viewer extension is a visualization tool for csv files. When the software user gets arrays of many rows and columns, it is essential to have a tool to visualize them since this IDE does not have it by default. The Python Docstring Generator extension was installed to provide the code with docstrings and descriptions of classes and functions.

Python version 3.8 was used for writing the code because of the compatibility with the OpenCV library. OpenCV is an open-source Computer Vision software library. OpenCV was built to provide a common infrastructure for Computer Vision applications and accelerate machine perception in commercial products. The library has more than 2500 optimized algorithms, which includes a set of both classic and state-of-the-art Computer Vision algorithms, e.g., visual tracking algorithms, camera calibration, and 3D reconstruction [82]. As one can see this is an essential library for the purpose of this master's thesis.

The OpenCV library was written originally in the C++ programming language. However, it can be used in Python code via the OpenCV-Python application programming interface (API). OpenCV-Python is a Python API for OpenCV that combines the best qualities of the OpenCV C++ API and the Python language.

Python virtual environments were used for the development of the software. This way, it was possible to access different configurations of Python packages. The essential packages, which were used in the context of this implementation are shown in the following list:

- numpy v1.20.3

- opencv-contrib-python v4.5.3.56
- pandas v1.3.1
- imutils v0.5.4
- matplotlib v3.4.2

NumPy is a Python library that supports improved computation with arrays and vectors compared to the standard Python library. Having NumPy installed is a requirement for using OpenCV. The selected OpenCV package was the extended one (contributors-package). It has some additional modules that the standard OpenCV package does not have, e.g., the tracker implementations. The pandas package was used for saving and reading csv files. The imutils package has some excellent functionalities when working with OpenCV, like, e.g., an implementation of a FPS counter that is used for measuring the processing speed of the algorithm. Finally, the matplotlib package was used for plotting and visualizing the results.

The methodology followed during the development was to write small scripts with the essential functions and test them. Afterward, new scripts were created integrating all the functions. The followed coding philosophy was function-oriented. This means that only functional programming was used. This has some advantages as one function can be used multiple times by only having to write one extra line in the code. In addition, a function can be loaded by other scripts so that the code looks cleaner than in other coding philosophies.

The essential scripts that were written are listed below.

- opencv_measurement.py
- camera_calibration.py
- pixel2metric.py
- plot_deflections.py

The target tracking algorithm was implemented in the script opencv_measurement.py. As a result, a csv file was saved so that the other scripts could read it and work with it. The csv file saves an array with twelve columns and many rows. In each row, the five target positions of the actual frame are saved. These are ten columns as the position are two-dimensional. The last two columns are the reference pixel coordinates where the original frame was cropped. In this way, the absolute deflections in the IPCS are obtained by adding the coordinates measured in the cropped frame to the reference value.

The camera calibration algorithm is implemented in camera_calibration.py. The calibration results are saved into binary files that the other scripts can read. The 3D reconstruction approach is implemented in the script pixel2metric.py. Here, the computation of the 3D coordinates from the 2D coordinates is done. In the script plot_deflections.py some nice plots were programmed for the visualization of the results.

Matlab was used for visualizing the results and for the creation of timetable variables from the csv files. The time-stamping algorithm needs the data to be in Matlab's timetable format.

Test and validation The software for wing shape measurement was tested both during and at the end of development. Most of the functionalities were individually tested before integration was carried out. In this way, a correct operation of the software after the integration of the functionalities was ensured.

This subparagraph describes the tests and presents the results that were obtained during the tests. In addition to the testing of the individual modules, pre-tests on some core functionalities were made.

Afterward, tests were also performed on the entire software. The software testing was carried out with videos from T-FLEX UAV test flights from the year 2019.

All the tests and pre-tests were made on a PC with the central processing unit (CPU) from Intel model Core i5-10600K [40]. This processor model has six cores and 12 threads. The clock of the processor runs at $4.10\ GHz$. The PC has $16\ Gb$ of random-access memory (RAM) and a graphics processing unit (GPU) from NVIDIA model GeForce RTX 3060 [16]. The OS of the computer is Windows 10 Professional [75]. The compilation number of the OS is 19043.1288. The system has a 64-bit architecture. Table 29 shows all the PC specifications described above.

Table 29: Specifications of the PC where the tests were performed

Component	Specification
CPU	Intel Core i5-10600K @$4.10\ GHz$
RAM	$16\ GB$
GPU	NVIDIA GeForce RTX 3060
OS	Windows 10 Professional compilation 19043.1288
System architecture	64-bit

The original video data of the Mobius cameras was in MOV format. The cameras are configured to save the videos in the SD-Card whenever the file size of the video reaches approximately $1GB$. This leads to the test flights being split into different video files with MOV format. Usually, a full test flight is split into three or four video files. The solution applied to obtain a video file of the full flight was to attach the multiple video files into one. This was accomplished by using the video editing software Camtasia 2020 [110]. After joining all video files together, the exported video file had an MP4 format. Both MP4- and MOV-format are encoded with the MPEG-4 codec, which is a standard codec in audiovisual files.

Pre-tests In this subparagraph, the performed pre-test and their results will be presented. Pre-tests on the stability and speed performance of the OpenCV-trackers were made. Thus, the wings were tested to check which were the best features to track. Camera calibration tests were also performed previous to the final testing of the software. Based on the results, the best performers were implemented in the final software.

As mentioned in subparagraph 2.3.3 the OpenCV library has a lot of implementations of Computer Vision functions. Tests to see the individual overall performance of these implementations were performed.

For the pre-tests of the OpenCV-trackers, only a fragment of a test flight was used. The video file is called *REC_0002.MOV*, and it was recorded in the test flight on the 1st of August 2019. The video has an original resolution of 1920x1080p but is cropped to a resolution of 1595x341p. In these pre-tests, two aspects were analyzed. The first is the tracker stability, i.e., how many times the tracker fails. Tracking failure is meant when the BB disappears or drifts away from the target that it is supposed to track. The second aspect is the speed performance of the tracker, i.e., how many FPS the tracker can process. A FPS counter was implemented in the software to obtain the mean value of the processed FPS of the video.

This pre-test was done three times at different targets of the wing: wingtip, approximately half of the semispan, and wing's root. The goal of the pre-test was to find the OpenCV-trackers that performed better in this setup. Tables 30 - 32 show the performance of the OpenCV-trackers in the two aspects mentioned above.

In tables 30 - 32, it is shown that the CSRT-tracker is the most stable implementation in OpenCV, followed by the MOOSE-tracker. The processing speed of the CSRT-tracker is a little bit lower than the average, but the reliability is the best. This characteristic makes the CSRT-tracker a good option for

Table 30: Pre-test of the OpenCV-trackers: wingtip

OpenCV-tracker	FPS processing speed	Tracking failures
CSRT-tracker	$\sim$ 27 FPS	2
KCF-tracker	$\sim$ 32 FPS	8
MOSSE-tracker	$\sim$ 45 FPS	4
BOOSTING-tracker	$\sim$ 32 FPS	12
MIL-tracker	$\sim$ 16 FPS	$>$ 30
TLD-tracker	$\sim$ 11 FPS	$>$ 30
MEDIANFLOW-tracker	$\sim$ 32 FPS	$>$ 30

Table 31: Pre-test of the OpenCV-trackers: half of semispan

OpenCV-tracker	FPS processing speed	Tracking failures
CSRT-tracker	$\sim$ 31 FPS	2
KCF-tracker	$\sim$ 34 FPS	6
MOSSE-tracker	$\sim$ 44 FPS	3
BOOSTING-tracker	$\sim$ 32 FPS	8
MIL-tracker	$\sim$ 18 FPS	$>$ 30
TLD-tracker	$\sim$ 14 FPS	$>$ 30
MEDIANFLOW-tracker	$\sim$ 33 FPS	$>$ 30

Table 32: Pre-test of the OpenCV-trackers: wing's root

OpenCV-tracker	FPS processing speed	Tracking failures
CSRT-tracker	$\sim$ 30 FPS	2
KCF-tracker	$\sim$ 32 FPS	6
MOSSE-tracker	$\sim$ 45 FPS	2
BOOSTING-tracker	$\sim$ 33 FPS	7
MIL-tracker	$\sim$ 19 FPS	$>$ 30
TLD-tracker	$\sim$ 10 FPS	$>$ 30
MEDIANFLOW-tracker	$\sim$ 31 FPS	$>$ 30

tracking the wing's outer targets. On the other hand, the best implementation in processing speed is the MOSSE-tracker. In addition, this implementation is very stable, matching the CSRT-tracker at the wing's root target. Thus, the MOOSE-tracker recovers on most occasions when a tracking failure happens. The BOOSTING-, MIL-, TLD- and MEDIANFLOW have so many tracking failures that they make them almost unusable. Thus, the processing speeds of the mentioned trackers are not outstanding.

This pre-test shows that the two best implementations for this scenario were the CSRT-tracker, where high stability is required, and the MOSSE-tracker for the inner wing targets, where stability is not a concern and pure speed is needed.

The goal of the next pre-test was to determine which targets should get tracked. For this purpose, the OpenCV function *cv2.goodFeaturesToTrack()* was used to determine strong corners on a frame. The function finds the most prominent corners in the image or in the specified image regions as described in [97]. Those prominent corners should match the best target locations on the wing. Three different background scenes were chosen to test which targets were the best for all background conditions. The same video as in the previous subparagraph was used for this pre-test. The maximum corner variable was set to five to detect the five most prominent corners. Figures 233 - 234 show different frames of a test flight, where the Shi-Tomasi Corner Detector is being applied.

In the figures above, one can observe that according to the Shi-Tomasi Corner Detector, the strongest

Figure 233: Shi-Tomasi Corner Detector under clouded background

Figure 234: Shi-Tomasi Corner Detector under green background (field)

corners are the wheel at the wingtip and the four servos under the wing. As mentioned above, with three different backgrounds, the corner detector finds the same five spots in all situations.

This pre-test shows that the best targets to track in the target tracking algorithm are the wheel at the wingtip and the four servos under the wing.

In the next pre-test, camera calibration of the two Mobius cameras was performed. The Mobius cameras were calibrated separately and marked with 'L' and 'R' to differentiate between the two cameras. The implemented calibration approach is based on the scientific paper by [125]. This approach is implemented in the OpenCV library in the function cv2.calibrateCamera().

A checkerboard pattern was printed and attached to a planar surface. 16 images were taken with each Mobius camera as the recommended minimum are approximately 10 images of the pattern [81]. The cameras were at a static position during the calibration. The checkerboard pattern was translated and rotated in each image, ensuring diversity in the images. The square side length was measured and is 27.6mm long. The calibration results are the intrinsic camera matrix, the coefficients of optical lens distortion, and the translation and rotation vectors. Note that the rotation vector can be converted into the rotation matrix by the Rodrigues formula [25]. This method is also implemented in OpenCV in the cv2.Rodrigues() function. During calibration, one can observe that the feature points were perfectly found. Thus, optical lens distortion is only visible when the checkerboard pattern is near the camera. This is not the case when the cameras aim at the FLEXOP wings, as there is a considerable distance between the camera and target points. The assumption of ignoring lens distortion that is made in [13] is also applicable in this scenario.

The results were saved as binary files in the npy-format (numpy). Rotation and translation vectors were not saved as the extrinsic parameters must be obtained with the camera mounted on the T-FLEX. The results of both Mobius camera calibration are shown in table 33 and 34.

Tests with videos of full flights In this chapter, the results with the full flight videos will be discussed. As mentioned above, full flight videos were obtained by attaching the different partial videos with the video editing software Camtasia 2020. In total, we obtained five complete videos of flight tests: three left-wing videos and two right-wing videos. The videos were edited so that the take-off phase starts

Table 33: Calibration results: left Mobius camera

Parameter	Value
focal length f_x	$1.46292577\,\exp(+03)$
focal length f_y	$1.46683821\,\exp(+03)$
Skew factor γ	0
Principal coordinate c_x	$1.02045320\,\exp(+03)$
Principal coordinate c_x	$5.88698589\,\exp(+02)$
Distortion coefficient k_1	-0.37839949
Distortion coefficient k_2	0.11026849
Distortion coefficient p_1	0.00342506
Distortion coefficient p_2	-0.001392
Distortion coefficient k_3	0.10850981

Table 34: Calibration results: right Mobius camera

Parameter	Value
focal length f_x	$1.50592831\,\exp(+03)$
focal length f_y	$1.50702372\,\exp(+03)$
Skew factor γ	0
Principal coordinate c_x	$1.03978361\,\exp(+03)$
Principal coordinate c_x	$5.30131041\,\exp(+02)$
Distortion coefficient k_1	-0.40766303
Distortion coefficient k_2	0.30293375
Distortion coefficient p_1	-0.0026379
Distortion coefficient p_2	-0.00163383
Distortion coefficient k_3	-0.20115594

only a few seconds after the video begins. The videos end a few seconds after the landing is completed and the FLEXOP has stopped. Table 35 shows the five videos and their following characteristics: video name, date, aimed wing, duration of the video, and weather conditions. The videos

Table 35: Calibration results: right Mobius camera

Video name	Date	Wing	Duration	Weather
190801_FT1_001_1_01_left.mp4	01.08.2019	Left	15:08 min	Partially clouded
191106_FT5_001_1_01_left.mp4	06.11.2019	Left	18:10 min	Very clouded
191106_FT5_001_1_01_right.mp4	06.11.2019	Right	18:14 min	Very clouded
191119_FT6_001_1_02_left.mp4	19.11.2019	Left	21:23 min	Clouded
191119_FT6_001_1_02_right.mp4	19.11.2019	Right	21:23 min	Clouded

The tracking failures in the full videos were tested. These tests were performed using the CSRT- and MOSSE tracker as these two showed the best performance in the pre-tests. The MOSSE-tracker was used for the two inner targets, and the CSRT-tracker was used to track the three outer targets. In this case, tracking failure means that the BB was not inside the AOI defined.This occurs primarily because of the BB disappearing or drifting away from the target. First, the CSRT-tracker stability will be analyzed in the five full videos to see how many times the tracker fails.

Analysis shows an acceptable tracker stability at the target 2. Taking in account the duration of the videos, it is a good performance because little intervention from the user is needed. In the second flight test plot, we can observe more tracking failures than in the rest of the videos. This could be because of this video's light conditions, which are very dark because of the very clouded sky. In addition, the plots

of the third flight test show no tracking failure in the right-wing and only one failure at the left-wing.

This is very similar at the location 3 because it is also the CSRT-tracker and the target is very similar. For the location 1 at the wingtip, the stability is not as good as location 2. This could be due to the wheel not being such a strong corner, added to the fact that this location has the fastest and biggest deflections. For both inner targets 4 and 5, many failures are shown in this plot. However, they do not fit the reality accurately. The MOSSE-tracker has many situations where it fails for one or two frames in a row but instantly recovers and continues tracking the correct target. This happens mostly when the video changes the predominant color, e.g., in fast maneuvers of the FLEXOP rapidly.

Another aspect is that the graph's pattern is very similar, comparing left-wing and right-wing. This is also positive because it gives credibility to our target tracking algorithm. The trend of the wing deflections is, in both cases, very similar.

Summary and outlook

In this subparagraph, the design and implementation of an in-flight wing deflection measurement system was presented. The state-of-the-art of science and technology in this field was presented. The three main fields of research are target tracking, camera calibration, and 3D reconstruction. After an overview of the topics, the theoretical foundations were explained. The critical mathematical concepts concerning this thesis were illustrated. Afterward, the developed algorithm for the wing deflection measurement was explained in detail. The development and implementation of the proposed system were presented. The main code was written in Python using the functions of the OpenCV library. Finally, pre-tests and tests and their results were discussed.

The tests showed that the best location for the target tracking was the wheel at the wingtip and the four servos under the T-FLEX wings. Testing also showed that the best OpenCV-trackers for the setup were the MOSSE-tracker and CSRT-tracker. The results showed a high-speed performance of the MOSSE-tracker and also good stability for the T-FLEX setup. The CSRT-tracker performed the best in stability terms. Also, the mentioned tracker performed nicely in the processing speed. The tests also showed good stability in the full flight videos. Locations 2 and 3 showed the best stability. In some of the analyzed videos, these two locations did not have any tracking failure.

A modification that could improve the proposed system would be to change the rear Mobius cameras for cameras with higher resolution and higher frame rate. This would lead to better colors, better contrast, and more fluidity in the recorded videos. These aspects could benefit the target tracking algorithm making it less prone to tracking failures. In addition, the software user could also benefit from these modifications because the target tracking algorithm would be even more automated and would not need so much reinitialization from the user.

Another improvement to the measurement system would be to add a 360-camera to record a 360-video of the test flights. In this work, research was carried out to mount a camera on the top of the fuselage.

This location was most appropriate because the black case under the FBG module can be easily modified and changed as it is 3D-printed. The idea was to extend the black case so that the 360-camera could sit on top. In addition, pre-tests were made to check if the location recorded both wings. The position met the requirements, so it was decided it would be the best location for the 360-camera.

The model of the camera is the Insta360 ONE X. A 3D model of the camera was made in CATIA V5 based on the camera blueprints published in the camera's webpage [39]. In addition to the 3D model, two other parts were designed in CATIA; the case of the Insta360 camera and the case's support. The following list shows the names of the designed 3D parts:

- *insta360_case_FLEXOP.CATPart*
- *insta360_ONE_X_camera.CATPart*

- *insta360_supoport_case_FLEXOP.CATPart*
- *insta360_support-case-camera_FLEXOP.CATProduct*
- *insta360_support-case_FLEXOP.CATProduct*

Adding a 360-camera to the system could improve the quality of the 3D reconstruction approach as stereo-vision methods often have better accuracy. On the other hand, the algorithm would become more complex as two images would need to be analyzed to do the 3D reconstruction. In addition, the 360-camera would need to be calibrated, increasing the complexity of the system. Calibrating 360-cameras can be more complicated due to the high optical lens distortion caused by ultra-wide-angle lenses. However, work has been done in the field of omnidirectional camera calibration like e.g., [92]. An implementation of the 360-cameras was not carried out due to not having enough time.

The calibration from camera to physical coordinate system was not achieved. It will be further investigated if the method is worth pursuing.

Testing the Flight Control Computer New IMU configuration

New IMU softwares are implemented based on the new concept. All the modification are backward compatible, so no need to worry if the FCC software's or the IMU software's version is different. In such cases, the reconfigured IMUs are working in the original operation mode.

The IMUs on the wing's leading edge (IMU No. 1, 2, 3, 7, 8, 9) give digital acceleration values in the x-, y- and z-direction and analog acceleration values in the z-direction only. All of these data are filtered with a high-pass filter. The IMUs on the wing's trailing edge (IMU No. 4, 5, 6, 10, 11, 12) provide gyro data around the axis x and axis y and both digital and analog acceleration values in the z-direction. All of these data are filtered with a high-pass filter. Data filtering is done with an IIR high-pass filter with a 0.1Hz corner frequency. It is used to filter out the offset error caused by the temperature. For better understanding, see the figure 235 and 236.

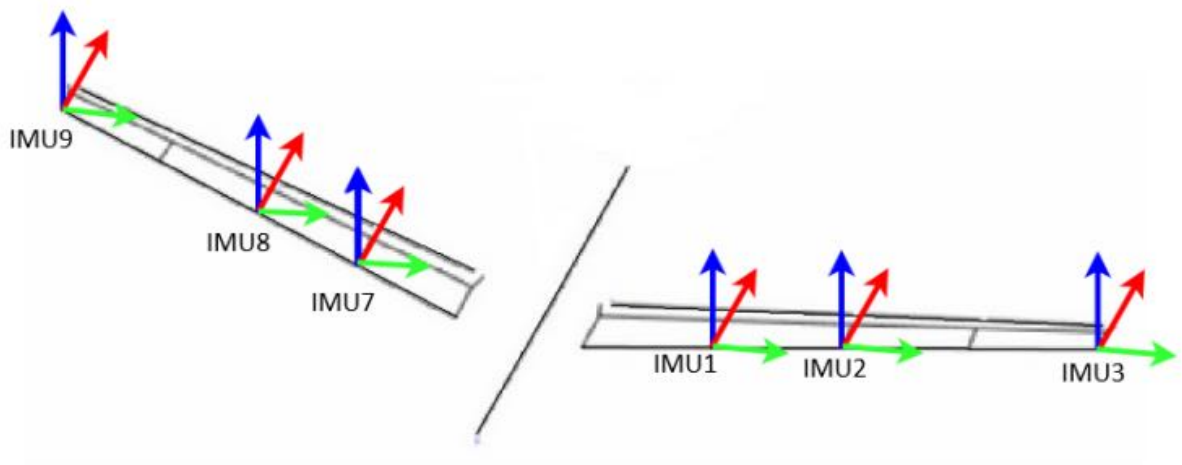


Figure 235: IMUs on the leading edge

Direct Drive

The actuator which responsible to move the 4th control surfaces during flutter control called Direct Drive. The actuator has an own controller, Flight Control Coputer just send position commands to it via CAN

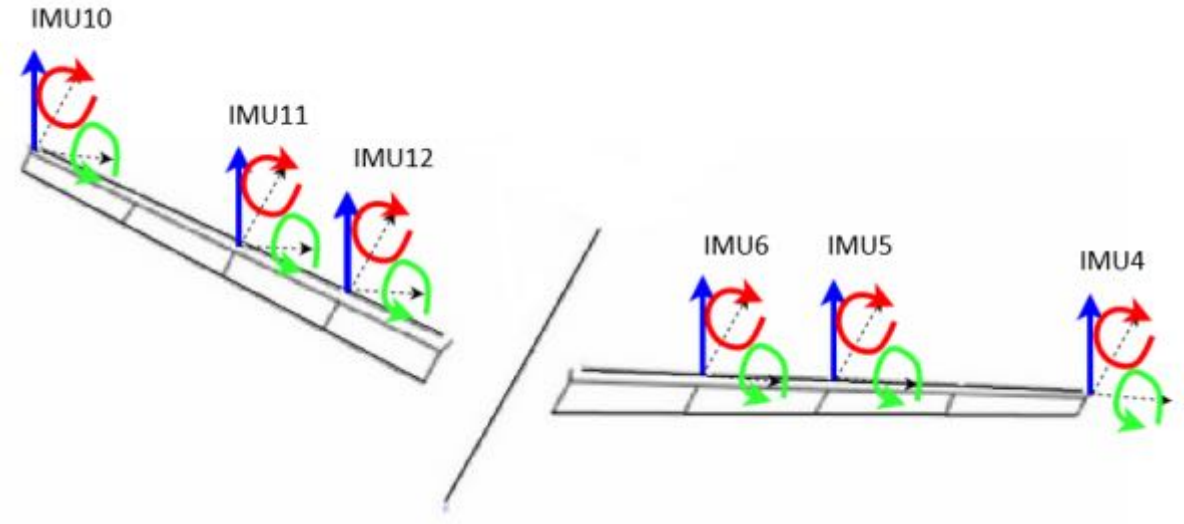


Figure 236: IMUs on the trailing edge

bus. Like all actuators, Direct Drive also connected to the RX-MUX-II. For compatibility with other wings which do not have Direct Drive just simple servos, the signal of 4th control surfaces will be sent throug the related PWM channels and in converted form through CAN bus as well. The figure 237 shows the route of the signal of the 4th actuator.

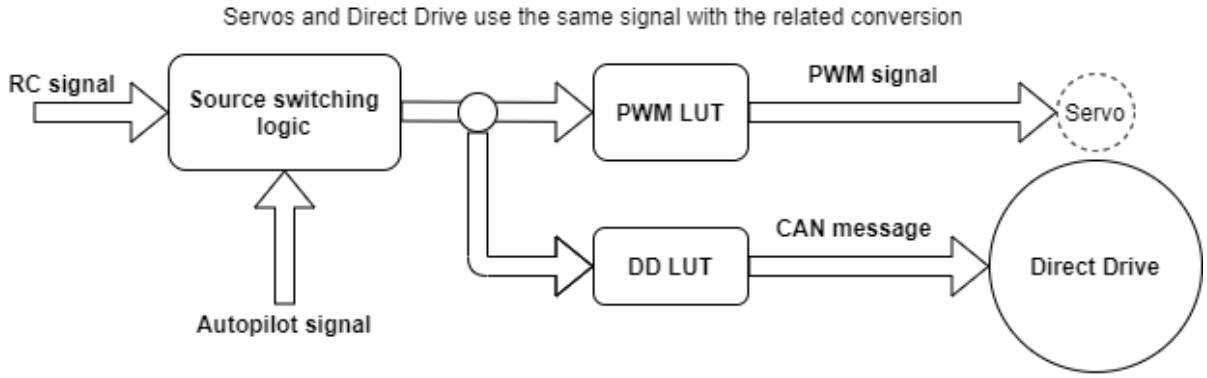


Figure 237: Signal route of the 4th actuator

Ground tests

To test the functionalities of the Flight Control Computer and its software with the autopilot before flight tests, we performed tests in Hardware-in-the-Loop test environment and on the real aircraft as well. To select the required autopilot functionality, we created a graphical interface which managed by the test engineer in the ground control station.

The main autopilot functionalities we tested:

1. Baseline functions

   (a) Autothrottle
   (b) Altitude holding
   (c) Course angle
   (d) Waypoint tracking

2. Identification functions

   (a) Signal injection to the engine
   (b) Signal injection to control surfaces

Hardware-in-the-Loop tests Baseline tests

For example, you can see how autothrottle test was performed in HIL. Figure 238 shows how throttle signal and airspeed changes if we give the following commands:

1. Use RC AP2 (augmented mode + autothrottle, nominal speed 38 m/s)
2. Reference velocity change from 38 m/s to 42 m/s
3. Reference velocity change from 42 m/s to 34 m/s
4. Reference velocity change from 34 m/s to 38 m/s

Identification tests

In engine identification mode, we inject step signals to the engine. Figure 239 shows how airspeed changes with an injected signal.

Throttle injection mode HIL test:

1. Mavlink in Baseline mode, augmented + throttle inject (open loop throttle in this SW_PI version) (velocity panel active)
2. 1st leg: start from RC AP1 trimmed 34 m/s straight and level (inner loop engaged), push 38 m/s button in Mavlink, switch to RC AP2, observe velocity increase with minimum pilot interference (RC throttle stick inactive) then switch back to RC AP1
3. 3rd leg: start from RC AP1 trimmed 34 m/s straight and level (inner loop engaged), push 42 m/s button in Mavlink, switch to RC AP2, observe velocity increase with minimum pilot interference (RC throttle stick inactive) then switch back to RC AP1

Signal injection mode HIL test:

1. Select Signal injection mode in Mavlink before flight and select between Flexible and Flight Mechanics tabs
2. Set initial velocity, amplitude multiplier and signal in RC AP1 during flight
3. Switch to RC AP2 to inject the selected signal

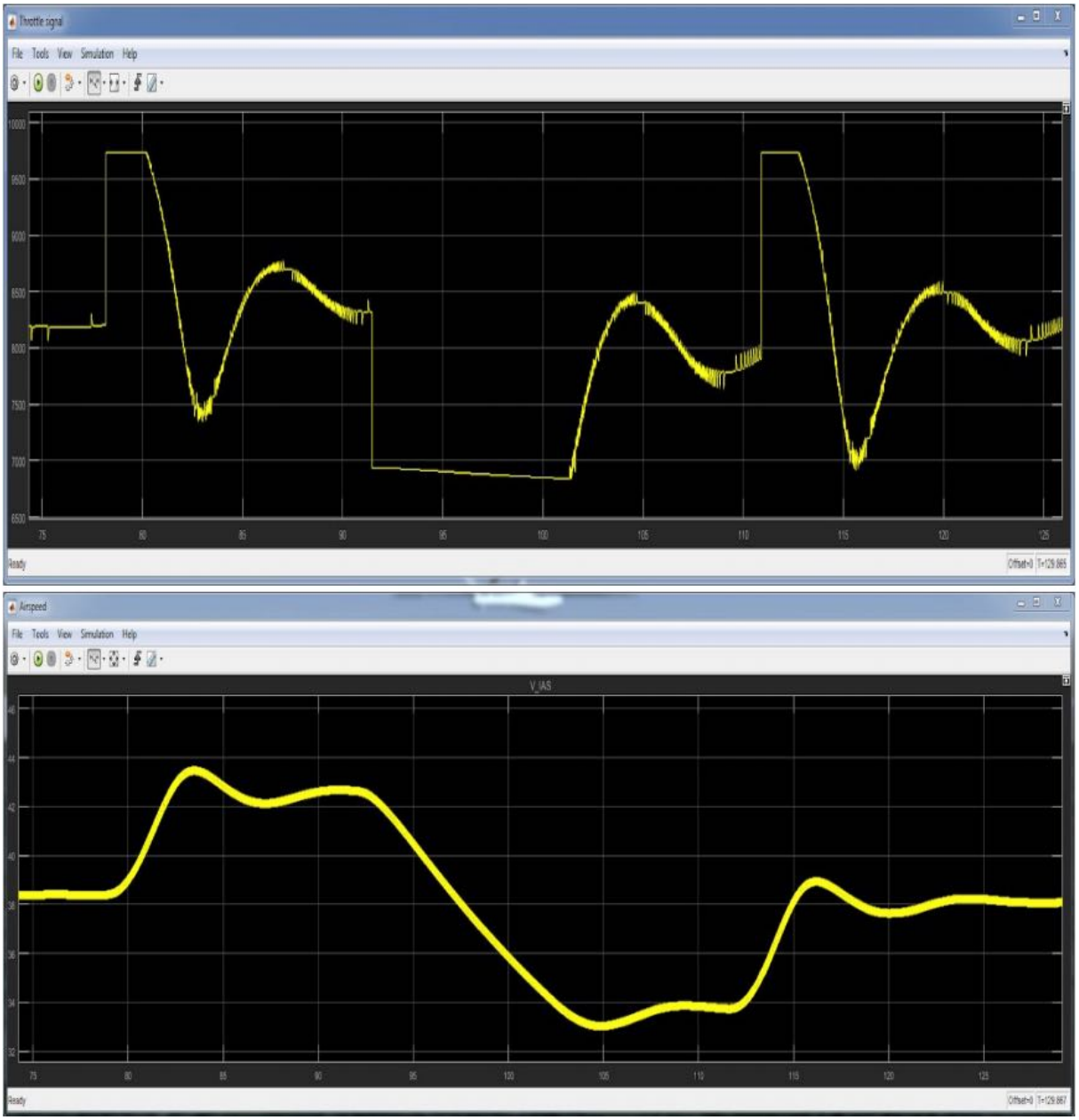


Figure 238: Autothrottle HIL test

Ground tests on the demonstrator aircraft

For ground tests we prepared a test version of autopilot with only one difference compared to flight version: we gave a constant value for the controller instead of measured speed.

Augmented mode test:

1. Moved the aircraft to change pitch and roll
2. Control surfaces tried to stabilize the aircraft

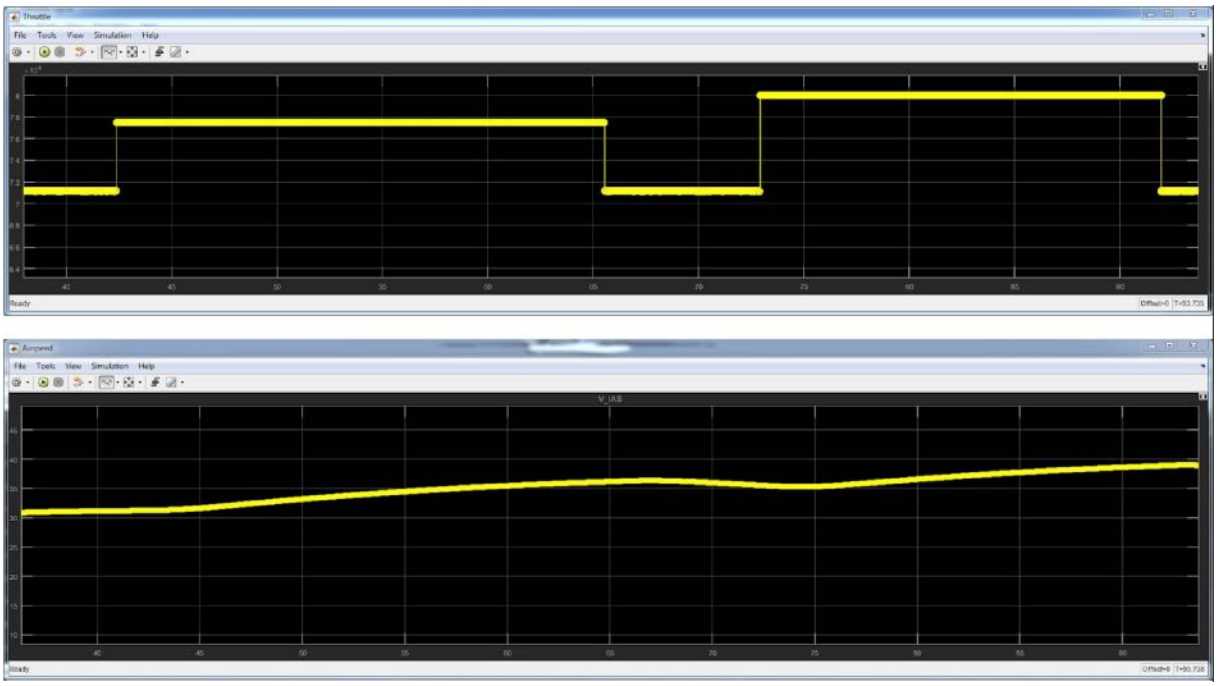


Figure 239: Throttle signal injection HIL test

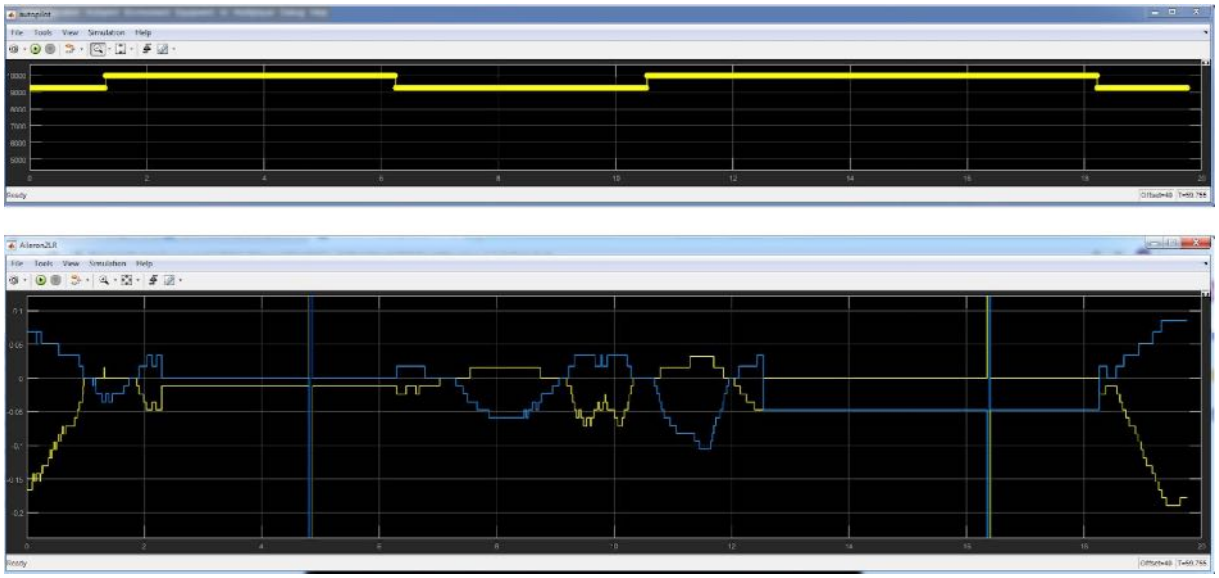


Figure 240: Signal injection test in HIL

Signal injection test:

1. Select initial velocity, amplitude multiplier and signal on mission planner (In RC AP1)
2. Inject the signal: switch to RC AP2
3. Related control surfaces moved according to the selected signal

Switching times [sample]:	Commanded velocity [m/s]:
373000	38
378100	42
379900	38
385700	34
390700	42
393600	34
405000	38
406700	42
409100	34
415300	42
417700	38
420600	34
423800	38

Table 36: The given velocity commands

Altitude and course angle test:

1. We gave altitude and course angle values via mavlink
2. Control surfaces tried to follow the given values

Autothrottle test:

1. We gave 38 m/s constant airspeed to the controller instead of measured value
2. Gave a velocity value via mavlink (in RC AP2)
3. The engine tried to increase or decrease the velocity depending on the given value

Table 36 shows the given commands and Figure 241 shows the response of the engine.

New test environment using Speedgoat target machine

During Harware In the Loop tests we faced with many problems, eg. model running was not rel time, caused by host os, there was no enough interfaces to test all functionalities of the Flight Control Computer, that is why we started to build a new HIL test environment based on a Speedgoat target machine. The target machine got many useful interfaces and it is capable to run our model real time.

The following table describes all interfaces and those purpose.

Implementation of the MATLAB HIL model of the aircraft on the new Speedgoat machine has begun. Currently it is in a state where the incoming CAN messages from the FCC are arriving via the IO612 card of the Speedgoat. Part of the Simulink block diagram which handles the CAN communication can be seen in Figure 242.

The upcoming task is to make the aircraft model (Simulink S-function) compatible with the OS of the Speedgoat. After that the necessary I/O for the model outputs must be implemented as well. These are serial ports that emulate the xSens and air data sensors.

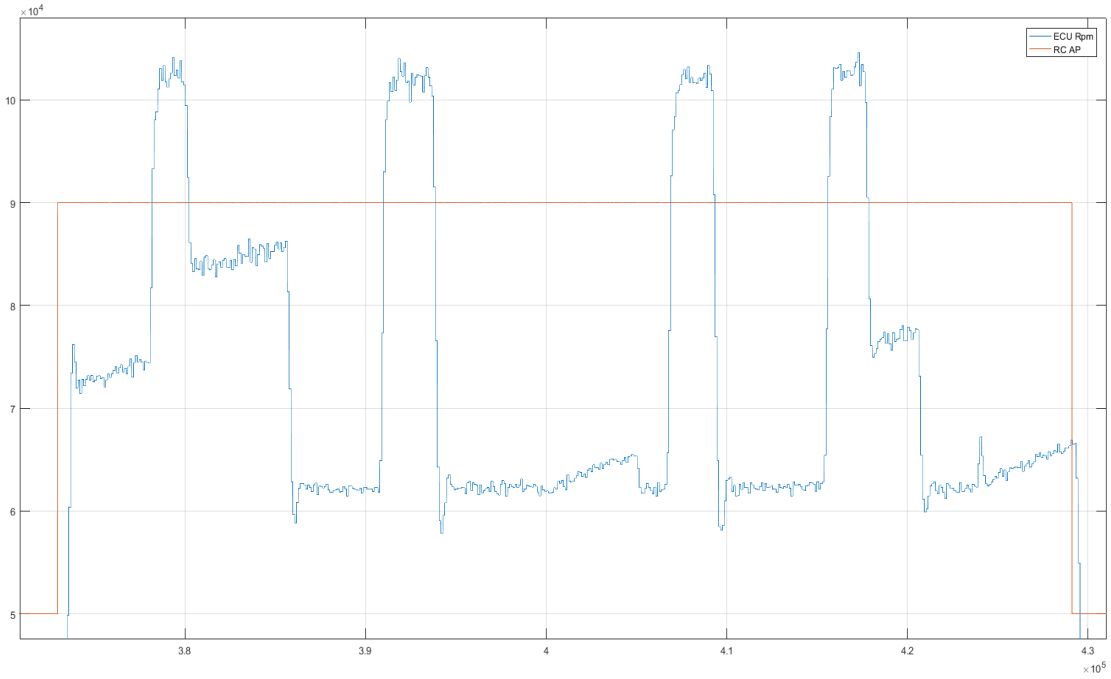


Figure 241: Engine response for the given commands

The Speedgoat target machine at DLR's facility is not equipped with a PWM capture card like the one at SZTAKI, around which we designed the HIL setup. To overcome this issue, we have revived and further developed the CAN-based solution from the old PC-based setup. In the old setup, the control surfaces were split into groups of four and then assigned a CAN arbitration ID. The groups were selected based on the physical PWM outputs on the old RX-MUX-II, but the CAN messages were sent out by the FlightHAT. For simplicity and because the FlightHAT is overwhelmed, we have chosen to send the messages from the RX-MUX-II. The old physical PWM output layout was long gone, if we had simply kept the old groups, we would have run into a problem. Most of the channels in the original groups were now separated between the two RX-MUX-II units. This meant that both RX-MUX-II units would have tried to send a part of the same message with the same CAN ID at around the same time. So, instead of just reorganizing the groups to match the current layout, we have come up with a new solution.

We have split the IDs, the last two digits represent the content of a message (i.e., which control surfaces

Interface	Amount	Purpose
RS-232	3	FlightHAT (ADS, Mavlink, Sindy/Fibre)
CAN HS	6	FlightHAT (IMU, SHM) RX-MUX-II (DD, new servos)
PWM	24	RX-MUX-II (actuators)
PPM capture	2	RC for SIL
UART (TTL)	3	ECU, RC for SIL

Table 37: Speedgoat interfaces

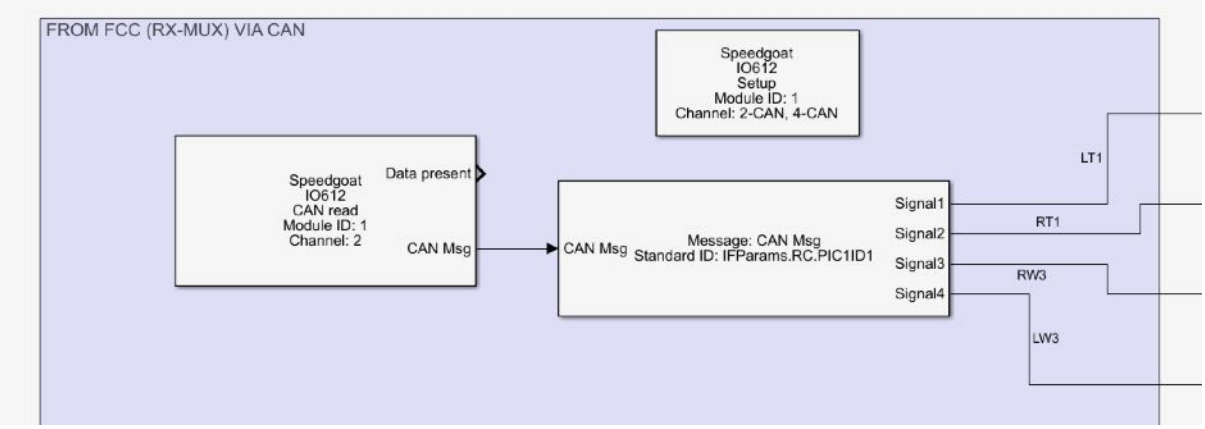


Figure 242: Speedgoat interface blocks

it represents), and the first two represent which RX-MUX-II sent them. So, for example, 1007_{10} means the second group (surfaces 4-7), sent from the first RX-MUX-II. The same message from the other RX-MUX-II would be 1507_{10}. The surfaces are numbered by the corresponding autopilot channel ID. If an RX-MUX-II is not responsible for one or more surfaces in a group, it replaces the value with a 0. The deflection of the surfaces is represented via the usual 5000-10000 range.

Remote Control System range test Integrated system at EDBC

There are two types of range-test conducted with the P-flex aircraft.

- Stationary test
- Taxi test

During the stationary test, the aircraft is in an elevated position about 1.5-2m from the ground, in a fully assembled state and most subsystems are functional. The RC controllers are with one or two operators, and they are driving away from the aircraft. Although the aircraft orientation is fixed relative to the remote controller, it would be possible to change the aircraft orientation as well.

It would be possible to map out different orientations of the aircraft relative to the remote controllers, and module it via different distances. However, this does not give a real implication of the maximum possible range of a given system, it can give a realistic safe distance.

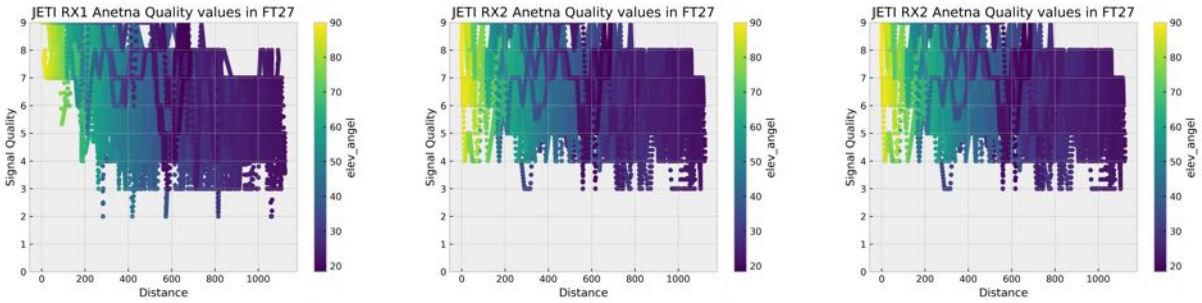
With our tests, the relative orientation of the aircraft was fixed into a not-ideal configuration, and we reached 700-800 metes with each RC system. These reseults meet the system requirements for the operational boundaries of the aircraft.

During normal taxi tests, the pilots drive the aircraft around and away from themselves, to validate the control ability of the system on the ground with different speed conditions. Conducting range-test while taxing, the pilots drive the aircraft slowly away from themselves in a controlled manner, normally up to 200-300 meters. The aim is to cover at least 2-3 times the normal takeoff distance on the ground, and demonstrate full control and ”acceptable” radio reception values during both directions and turning.

During our taxi range test, the RC system successfully passed the above-mentioned taxi test.

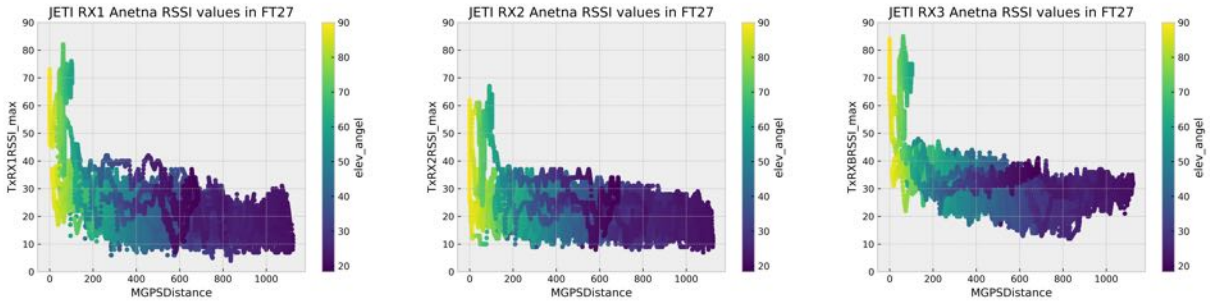
Furthermore, the backup remote control system reception characteristics were checked after each flight, to have a better understanding of the normal behaviour of the system. Figure 243 and 244 show the

different measurement point distributions over distance.



(a) RX1 **(b)** RX2 **(c)** RX3

Figure 243: JETI Antenna values by internal indicator, logged on the backup system.



(a) RX1 **(b)** RX2 **(c)** RX3

Figure 244: JETI RSSI values logged on the backup system.

Telemetry system range test As mentioned in [111], the telemetry system received major changes during the rebuild. As with any RC aircraft, it needed to be tuned and changed during operation, to reach an acceptable quality.

However, during the first flight-test campaign, significantly bad reception and data quality was experienced with both telemetry links. By changing the internal configuration of the 868Mhz module, and applying an additional power supply to it on the GCS side, the system behaviour became stable and usable on the 868Mhz data link.

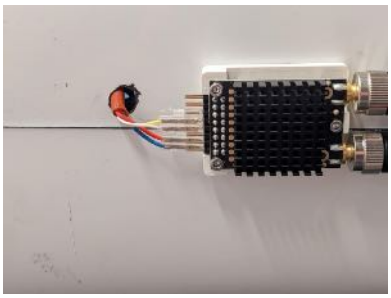
Until the end of the first campaign, it was possible to reach data link quality similar to that observed during initial telemetry range testing with the clean mockup configuration [123].

After the first campaign, the 433 Mhz telemetry system was replaced by an 868 Mhz system. To avoid interference, the MisiPlanner(MP) telemetry module was now located under the belly of the aft of the landing gear. With that, potential interference between the three 868 MHz radio systems is minimized. The main reasoning behind the change, and location:

- The 433Mhz telemetry setup had slightly worse reception than in the previous configuration. The quality was highly dependent on the relative orientation of the plate to the Ground Control Station(GCS).

- The 868Mhz had acceptable reception, independent of the orientation of the plane

(a) 868Mhz telemetry module location for the MP



(b) 6 pin layout cabling for the telemetry module

Furthermore, the following changes were made during the flight test campaign in May 2023:

- Dedicated power supply modes were fabricated from a 5V power supply unit for the telemetry modules in the GCS. It was found that the USB cable extenders were not supplying enough power to the telemetry modules in the GCS, although both extension cables had dedicated power supply modules.
- The channel, frequency and communication speed were changed on each modem pair.
  - The usable frequencies were put to the two halves of the available spectrum, to eliminate possible interference.
  - Data communication speed base-frequencies were removed to the bare minimum needed to support the data stream. This normally helps eliminate package and data errors.
- The frequency and channel layout of the long-range communication system for the pilots were changed. It was found that they were interfering with each other and with the telemetry system as well.

During the final flight week, an unidentified error in the EDL telemetry stream was however still observed. At seemingly random times, the datastream stopped for 30 seconds. It was possible to identify, that the connection loss has nothing to do with orientation, distance, power supply or radio interference, so likely it is a software-related error. Since it was not possible to reliably reproduce the data loss, it was also not possible to fix the problem.

Static test of -1 wing This chapter presents the airworthiness test performed on the -1 wing to verify its capacity to withstand the design load of 4 g. It encompasses an explanation of the arrangements made for both the hardware and simulation aspects, along with specific information about the execution of the test and its outcomes.

Preparation

During the airworthiness test, the wing's entire surface cannot be distributedly loaded as it would be in flight. Instead, sandbags are strategically positioned at specific locations on the wing to simulate the distributed load conditions experienced during flight. To ensure an effective distribution of the load and account for the wing skin's load-bearing capacity, the load application sections are chosen to align with the rib locations. Consequently, there are seven sections on each side of the wing, and there is no additional space available to accommodate extra sandbags. To determine the optimal weight of

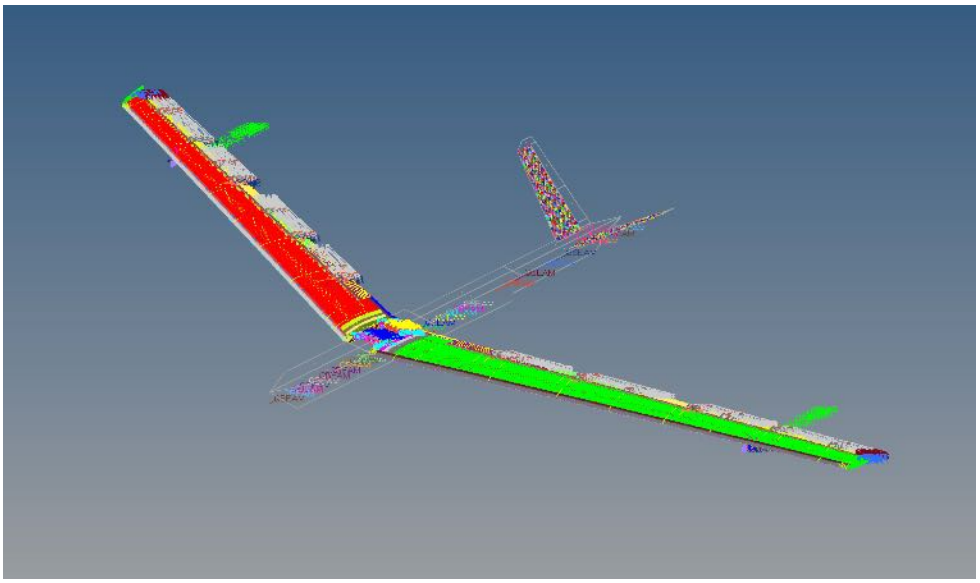


Figure 246: -1 wing FEM in trim analysis.

sandbags needed to accurately replicate the flight configuration, a simulation study must be conducted. The simulation model utilized for this purpose is illustrated in Figure 246.

The first step of the simulation study involves performing a trim analysis to ascertain the aerodynamic loads acting on the aircraft and determine the flight shape of the wing. This trim analysis relies on the Nastran solution 144. The displacement obtained from the trim analysis is subsequently utilized as constraints in the static analysis to calculate the section loads (Figure 247).

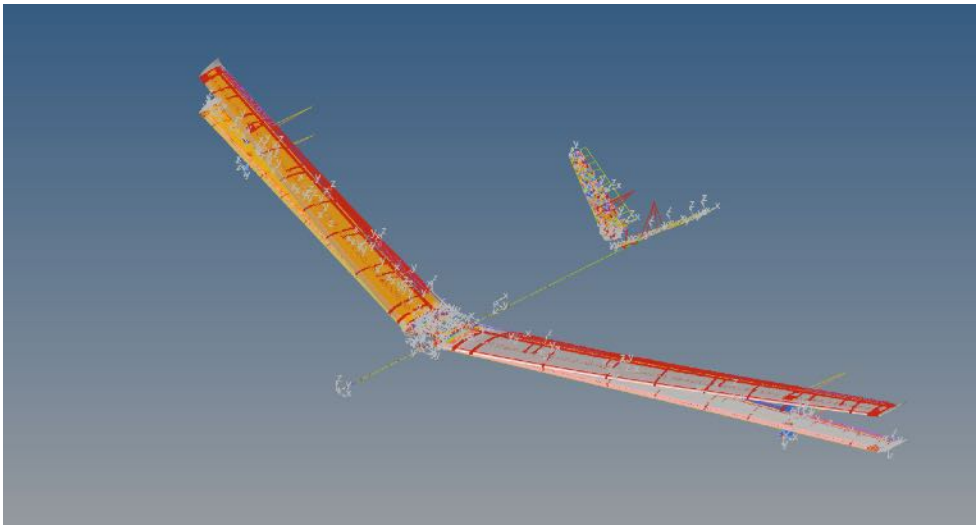


Figure 247: -1 wing static analysis.

The resulting loads and positions for each section are presented in Table 38. The sandbags are prepared according to the specified weights provided in Table 38.

Conduct

Table 38: Sectional loads on -1 wing.

Section	Section force (N)	Sandbag weights (kg)	Distance (m)
1	86,20	8,79	0.5
2	164,34	16,75	0,87
3	112,15	11,43	1,39
4	98,79	10,07	1,79
5	67,57	6,89	2,20
6	46,96	4,79	2,54
7	23,97	2,44	3,38

In order to ensure stability during the testing process, a wing stand was constructed and securely fastened to the ground. To replicate partial lift, the wing was positioned upside down, utilizing the force of gravity. Instead of suspending the sandbags from the wing, they were placed directly on the wing surface. This approach increased the contact area and reduced localized pressure on the fragile -1 wing skin, addressing concerns regarding potential damage. Given the high flexibility of the -1 wing, subjecting it to a 4g load could lead to substantial deflection, causing the sandbags to slide off. In a worst-case scenario, the wing could flip over and suffer damage. To prevent such issues, a rubber mat was affixed to the wing and positioned beneath the sandbags. The entire setup can be seen in Figure 248.

Figure 248: -1 wing airworthiness test.

Before applying the load to the wing, the distance between the wing tip and the ground was measured using a tape measure. Subsequently, the wing was loaded progressively from the inner section towards the outer section. Once the wing was fully loaded with a 4g force, the distance between the wing tip and the ground was measured again.

Results The conclusion drawn from the airworthiness test conducted on the -1 wing is that the wing is capable of withstanding the specified load. According to the simulation results, the deflection observed under the 4g load is measured to be 0.32 m. However, when measured using a tape, the actual deflection is recorded as 0.24 m. It is important to note that the measurement of deflection with a tape may introduce some deviation in the results. Therefore, it is possible that part of the difference between the simulated and measured deflection values can be attributed to the measurement method itself.

GVT - Introduction Ground Vibration Testing (GVT) is used to evaluate the structural dynamic behaviour of aircraft. During a GVT, the aircraft is suspended from bungees and subjected to controlled vibrations generated by electrodynamic shakers. The aircraft response to these vibrations can be used to identify resonance frequencies, damping ratios and mode shapes on ground before first flight of a prototype. The results are used to validate and update the Finite Element (FE) model for further aeroelastic simulations. The GVT is an important part of the development process for new aircraft and is performed at various stages of the design.

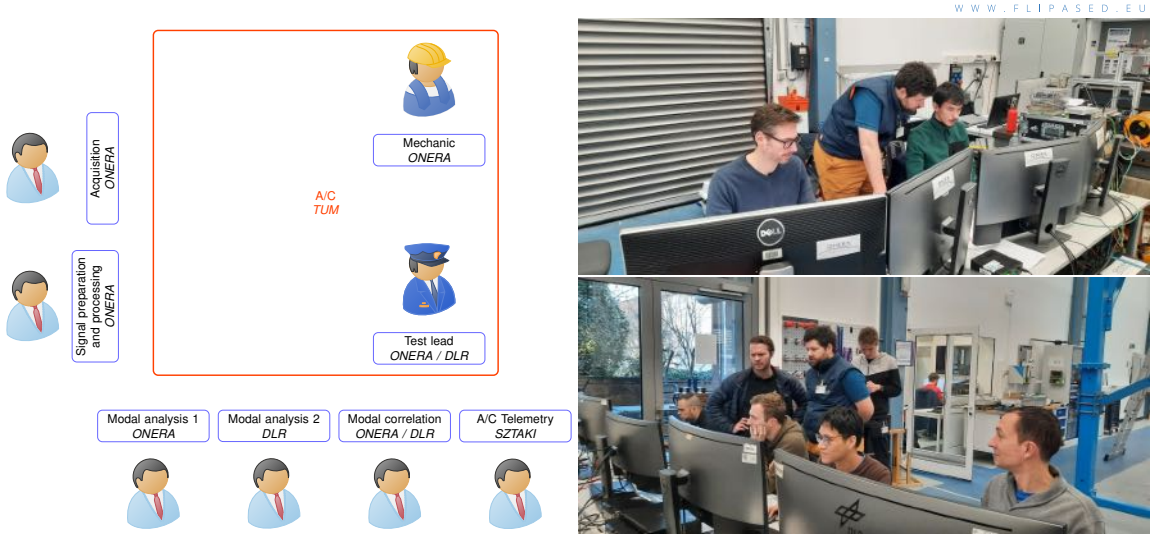
GVT - Test Overview The GVT of the FLIPASED aircraft was conducted by a team from ONERA and DLR at the DLR Institute of Aeroelasticity in Göttingen. The aircraft was suspended from soft bungee cords as shown in figure 249 in order to separate the elastic modes from the rigid body modes. The

Figure 249: FLIPASED UAV suspended from bungees in lab.

aircraft was then instrumented with approximately 150 lightweight accelerometers including uni-axial and tri-axial sensors. These provide the benefit of not mass loading the structure, while maintaining high accuracy even at low frequencies. Several electro-dynamic shakers were used to excite the structure with different custom designed input signals. A specialized test team consisting of a team leader, and engineers and computer scientists for data acquisition, signal processing, modal analysis and modal model correlation ensure high quality results, a plan of the work stations is depicted in figure 250.

GVT - Test Plan The GVT described in this document is the second performed on this aircraft. The first test was carried out during the EU-FLEXOP project by DLR on the T-FLEX demonstrator [113]. The current GVT has been dedicated to the thorough analysis of the -1 wing set of the aircraft, as well as the new P-FLEX airframe given the crash of the T-FLEX just weeks before the initial planned GVT date. Given the expected unstable behavior of the aircraft equipped with this wing set, an extensive test plan was performed on the aircraft.

Three structural configurations were tested during this GVT campaign



(a) Schematic organization with roles and *partner*.

(b) Pictures made during the GVT.

Figure 250: Test work stations.

- C1a : aircraft with flutter stopper masses in aft position (flutter-prone design point). This is the main configuration that was tested
- C1b : aircraft with flutter stopper masses in forward position (flutter-free design point). Three excitation runs were performed, all on the wings
- C1d : aircraft with flutter stopper rods removed. This configuration was not initially planned. It was performed as a fast verification point for the test instrumentation with a simpler aircraft structure.

GVT - Experimental setup Aircraft configuration

During the test, the aircraft weighed 70 kg which included 10 kg of fuel load. During measurements, the canopy was closed, and additional masses were fitted onto the airframe in order to account for missing components that would be present during the flight test campaign. At the beginning of the test, all control surfaces (Flaps 1 to 3 and ailerons 4) on both sides of the aircraft were powered on and flight ready. The ailerons were subsequently powered off due to high electromagnetic noise levels.

Measurement setup

The aircraft instrumentation setup is illustrated in figure 251. The list of sensors and nomenclature used is detailed in table 39. The aircraft was equipped with sensors on all main structural components and control surfaces, including wings, flutter stopper rods, v-tail, engine, fuselage and aero boom. The suspension bungees and supporting structure (sensors not represented or listed) were instrumented as well, which proved useful during the correlation of the modal data. The only massive component that was not instrumented is the landing gear which was considered relatively stiff. All sensors were positioned tangentially to the local surface.

The measurement setup was composed of a mixed instrumentation between ONERA and DLR hardware :

- ONERA: Wing, flutter stopper assemblies, and V-tail tri-axis acelerometers, shakers and related hardware (cabling, acquisition system, power amplifiers)

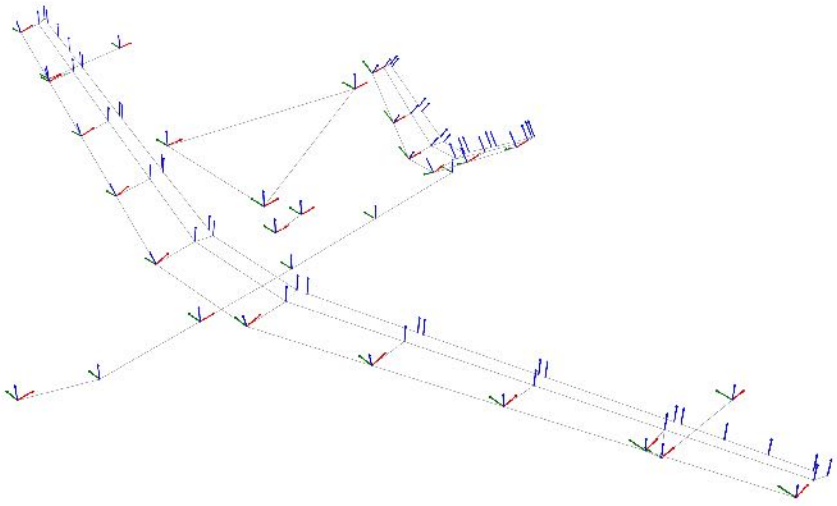


Figure 251: Aircraft sensor locations and directions. Local X (→), Y (→), Z (→) sensor directions.

- DLR: Nose boom, fuselage and engine tri-axis sensors, all uni-axis sensors (very light sensors), related hardware (cabling, acquisition system)

Sztaki provided a ground control station with telemetry capabilities to control the aircraft state and record data using internal sensors for other partners.

Excitation points

5 excitation points were exploited on this aircraft :

- Inner wing Z excitation
- Inner wing X excitation
- Fuselage Y excitation
- Fuselage Z excitation

All excitation points were glued onto the aircraft structure, via interface parts. The wing X excitation points were glued through a 3D-printed conformal interface part.

Suspension

The aircraft was suspended using three bungees and slings, two fitted on the wing roots on each side of the fuselage, and one under the fuselage close to a fuselage web location. The suspension was equipped with tri-axis accelerometers, to avoid any miss interpretation between the suspension chord modes and the aircraft flexible modes. The suspension gantry was equipped with accelerometers as well, for surveillance purposes, and weighted to ensure its stability.

Excitation strategy

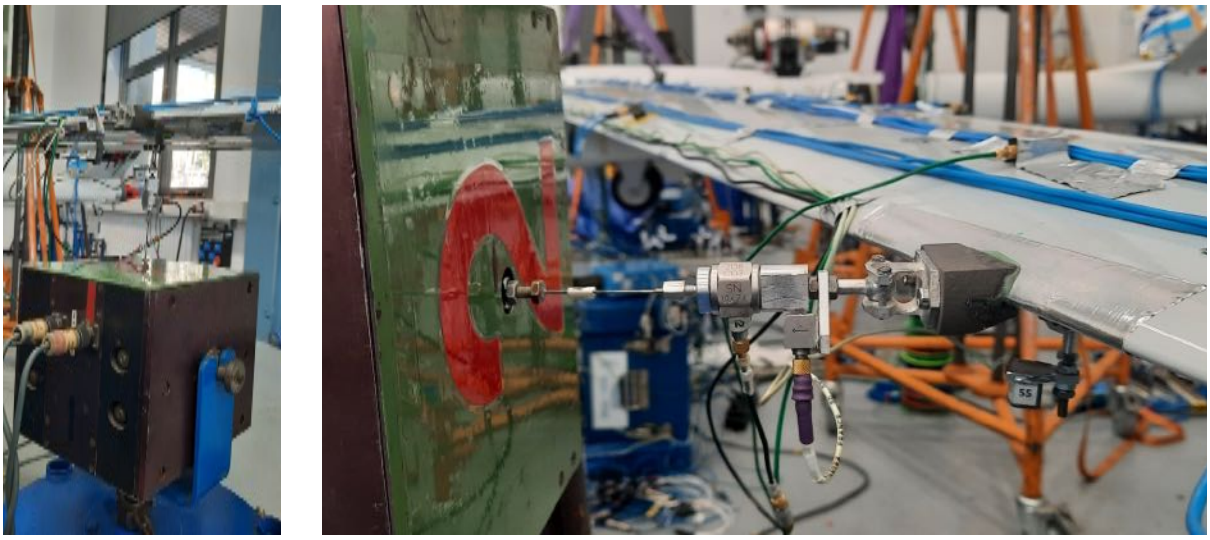
Component	Abbreviation	Number	Sensors	Channels
Left Wing	WNGL	01	6 XYZ + 7 Z	38
Left Aileron 1	AILL1	03	2 Z	
Left Aileron 2	AILL2	05	2 Z	
Left Aileron 3	AILL3	07	2 Z	
Left Aileron 4	AILL4	09	4 Z	
Left Direct Drive	DDL	17	1 XYZ	
Right Wing	WNGR	02	6 XYZ + 7 Z	38
Right Aileron 1	AILR1	04	2 Z	
Right Aileron 2	AILR2	06	2 Z	
Right Aileron 3	AILR3	08	2 Z	
Right Aileron 4	AILR4	10	4 Z	
Right Direct Drive	DDR	18	1 XYZ	
Left Stabilizer	VTLL	11	3 XYZ + 5Z	18
Left Elevator 1	ELEL1	13	2 Z	
Left Elevator 2	ELEL2	15	2 Z	
Right Stabilizer	VTLR	12	3 XYZ + 5Z	18
Right Elevator 1	ELER1	14	2 Z	
Right Elevator 2	ELER2	16	2 Z	
Fuselage	FUSE	19	1 XYZ + 4YZ	35
Engine	ENG	20	2 XYZ	
Nose Boom	NBOOM	21	1 XYZ	
Driving Points	DP	80	2 X + 1Y + 3 Z	
Suspension Device	SUSP	30	3 XYZ	

Table 39: List of sensors per aircraft structural components.

(a) Rearview, Y direction excitation point.

(b) Side view, Z direction excitation point.

Figure 252: Fuselage excitation points, equipped with a 47 N ONERA shaker along with its sensing elements.



(a) Z direction.

(b) X direction.

Figure 253: Wing excitation points.

Most of the analyses were performed using the *Phase Separation Method* (PSM) under random or swept sine excitations.

GVT - Test Results General comments

Rigid body modes of the aircraft identified during this campaign were not extensively identified. While the frequency and damping factors may be well identified, the generalized masses for these modes must be taken with precaution. This can be explained by the usage of lightweight sensors during this GVT to reduce perturbation of the aircraft structural behavior by the additional sensor masses. In fact, these sensors show poor sensitivity at low frequencies for technological reasons, leading to ill-identified generalized masses. Finally, the rigid body roll mode could not be identified during this test, due to its very low frequency.

Some flexible mode modal masses also suffer from poor identification, in particular, the one of mode n°12 `flutterstop_bend_y-a` identified in C1b. While there is no obvious clue as to why this mode suffers from poor identification, the fact that it is a flutter stopper mode could indicate and influence of the flutter mass inside the flutter stopper rod. In fact there is some freeplay between the mass and the rod, potentially resulting in non linear phenomena such as friction and impacts.

As a final comment, the frequency, and therefore the damping and modal masses of a few modes, in particular the in-plane or so-called scissor mode, was without surprise difficult to ascertain. This particular mode stems from the flexibility of the linkage between the aircraft fuselage and wings. On this aircraft, this linkage is bolted. Such configurations are often prone to non-linear behavior due to the high sensitivity of local sliding or even separation between initially in contact components. This leads to complex structural behavior on which linear modal identification reaches its limits. This is generally true for most linkage related flexibilities. As an example, figure 254 displays the frequency response functions of the wing X direction accelerometers, in C1a, during three fuselage Y excitation runs at increasing excitation force levels. By comparing the frequency response function trends around 8.5 Hz, 13 Hz and 27 Hz, one can observe a frequency shift of the response amplifications towards lower frequencies, as well as variation of these amplifications. These observations are easily attributed to sliding phenomena in the aircraft structure. Given the identified modal data in table 40, one can

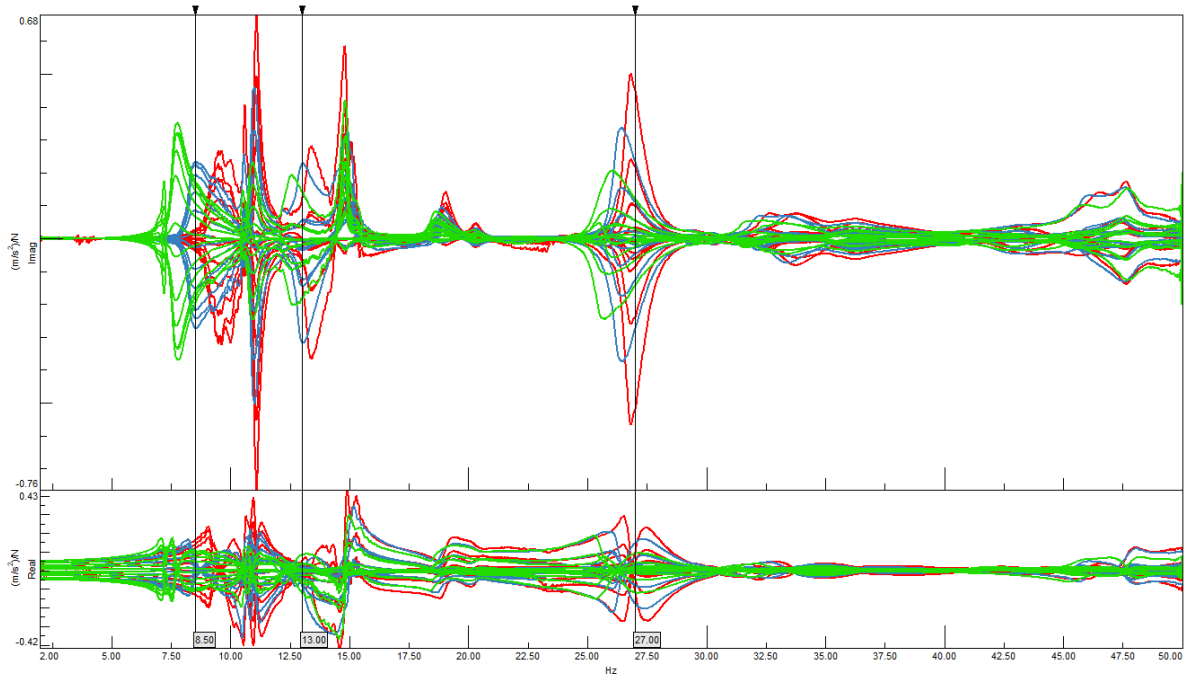


Figure 254: Frequency response functions of wing X direction accelerometer responses under fuselage Y excitation. Force voltage 0.25 V (—), 0.5 V (—), 1 V (—).

link these nonlinear behavior with the scissor/inplane mode, the V-tail rock mode, and the first lateral fuselage bending.

C1a

C1a modal results are gathered in table 40.

C1b

C1a modal results are gathered in table 41.

C1d

C1a modal results are gathered in table 42.

N°	Name	Damped Freq. [Hz]	Damping [%]	Gen. Mass [kg.m^2]	Norm. Point	MIF
1	ac_y_trans-S	0.867	0.83	284.99	8001905+Y	891
2	ac_pitch-S	0.888	5.14	12.57	2100101+Z	833
3	ac_x_trans-S	1.009	1.69	72.83	8000204+X	740
4	ac_z_trans-S	1.143	4.61	22.77	8001905+Z	928
5	ac_yaw-A	1.797	0.77	44.86	0100701+X	951
6	2n_wing_bend-s	2.938	1.10	5.08	0100704+Z	967
7	3n_wing_bend-a	7.220	0.79	3.24	0200704+Z	986
8	1n_wing_inplane-a	8.491	1.83	32.43	8001905+Y	898
9	wing_tors-s	10.744	0.95	0.65	0100405+Z	907
10	wing_tors-a	11.155	1.07	0.69	0200405+Z	930
11	4n_bending-s	12.023	0.72	3.12	0100704+Z	907
12	vtail_rock-a	12.501	3.36	0.64	1100304+Z	985
13	2n_wing_inplane-s	14.846	1.19	2.34	0100405+Z	960
14	flutterstop_right_y	16.702	2.97	0.44	0200405+Y	949
15	mass_lat-a	18.617	1.08	1.83	0200405+Y	847
16	mass_lat-s	19.298	1.41	6.85	0200405+Y	860
17	5n_wing_bend-a	20.383	1.78	1.57	0100704+Z	867
18	flutterstop_left_y	22.213	4.01	0.47	0100405+Y	987
19	2n_fus_vert-s	23.986	1.25	1.21	1200304+Z	972
20	6n_wing_bend-s	25.860	1.82	0.97	1200304+Z	922
21	2n_fus_lat-a	26.266	1.42	1.76	2100101+Y	902
22	2n_tail_bend-s	27.218	0.90	1.18	2100101+Z	779
23	7n_wing_bend-A	29.431	1.37	2.09	2100101+Y	888
24	2n_tail_bend-s?	29.466	1.10	1.40	2100101+Z	786
25	engine_x-s	32.172	0.74	2.86	2100101+Y	889
26	7n_wing_bend-A???	32.864	1.15	0.92	2100101+Y	863
27	2nd_wing_tors-a	33.350	3.45	0.74	2100101+Y	833

Table 40: Modal property table for FLIPASED – C1a.

N°	Name	Damped Freq. [Hz]	Damping [%]	Gen. Mass [kg.m^2]	Norm. Point	MIF
1	ac_pitch-s	1.309	4.78	25.89	1900501+Z	908
2	2n_wing_bend-s	2.952	1.35	4.86	0200704+Z	970
3	3n_wing_bend-a	7.327	0.76	3.07	0100704+Z	981
4	1n_wing_inplane-a	10.913	2.59	2.92	1200304+Z	985
5	4n_wing_bend-s	12.042	0.94	1.98	0100704+Z	955
6	2n_wing_torsion-s	12.756	0.71	0.57	0200405+Z	970
7	1n_wing_torsion-a	13.491	1.58	1.66	1100304+Z	844
8	vtail_rock-a	14.065	1.98	0.90	0200405+Z	632
9	2n_wing_inplane-s	16.005	0.61	4.46	0200701+X	966
10	5n_wing_bend-a	20.145	1.87	1.06	0200704+Z	977
11	2n_fus_vert-s	23.908	1.14	2.66	1200104+Z	953
12	flutterstop_bend_y-a	25.037	1.07	1148.28	0200405+Z	489
13	flutterstop_bend_y-s	25.396	1.24	3.83	1100304+Z	720
14	2n_tail_bend-s	25.984	1.67	1.44	1600304+Z	915
15	2n_fus_lat-a	26.686	1.60	0.57	2100101+Y	940
16	boom_bend_z-s	29.160	1.63	0.33	2100101+Z	761
17	boom_bend_y-a	30.219	1.41	0.08	2100101+Y	970
18	boom_bend_yz-a	32.469	1.55	0.12	2100101+Z	940
19	7n_wing_bend-a	35.580	2.48	11.08	0100405+Z	884
20	flutterstop_bend_z-a	39.249	3.36	0.30	0700304+Z	848
21	flutterstop_bend_z-s	42.313	2.61	2.41	0200405+Z	859

Table 41: Modal property table for FLIPASED – C1b.

N°	Name	Damped Freq. [Hz]	Damping [%]	Gen. Mass [kg.m^2]	Norm. Point	MIF
1	ac_x_trans-S	1.028	1.50	14.32	0100201+X	910
2	ac_pitch-S	1.353	6.77	20.70	1200101+Z	966
3	2n_wing_bend-s	3.081	1.18	4.89	1000704+Z	985
4	3n_wing_bend-a	7.491	0.77	3.00	1000704+Z	991
5	1n_wing_inplane-a	10.721	4.18	17.42	1100201+Z	962
6	4n_wing_bend-s	12.097	0.65	12.04	8000203+Z	928
7	vtail_rock-a	13.747	3.28	0.82	1200304+Z	991
8	2n_wing_inplane-s	15.369	0.82	16.09	0200401+X	981
9	5n_wing_bend-a	20.311	1.73	5.93	0200701+Z	831
10	2n_wing_torsion-s	22.808	1.03	3.08	0100702+Z	892
11	1n_wing_torsion-a	23.601	1.28	2.71	1000404+Z	960
12	vtail_bend-s	25.087	1.56	1.32	1100301+Z	954
13	2n_fus_lat-a	27.418	1.76	1.12	2100101+Y	862
14	6n_wing_bend-s	27.940	1.56	14.61	1600204+Z	816
15	2n_fus_vert_s	29.559	1.66	13.70	0100701+Z	742
16	boom_bend_lat-a	30.481	1.02	0.19	2100101+Y	953
17	boom_bend_vert-s	32.291	2.00	0.33	2100101+Z	870
18	7n_wing_bend-a	33.111	1.51	36.98	1000404+Z	596
19	3n_wing_torsion-a	34.959	1.10	2.89	3000101+X	535

Table 42: Modal property table for FLIPASED – C1d.

Comparison between configurations

Given the specific design point of the -1 wing set of this aircraft, with flutter masses designed to allow transitioning from flutter–prone to flutter–free behavior, it was of utmost importance to assess the actual effect of these masses on the aircraft behavior.

A list of common mode shapes identified in the three structural configurations mentioned in 2.3.3 was assembled and their frequency and damping properties were compared. As a recall, only configuration C1a was extensively analysed, and in C1b and C1d only wing excitations were performed.

Figure 255 depicts the qualitative orderding of modal families in each configuration. Unsurprisingly,

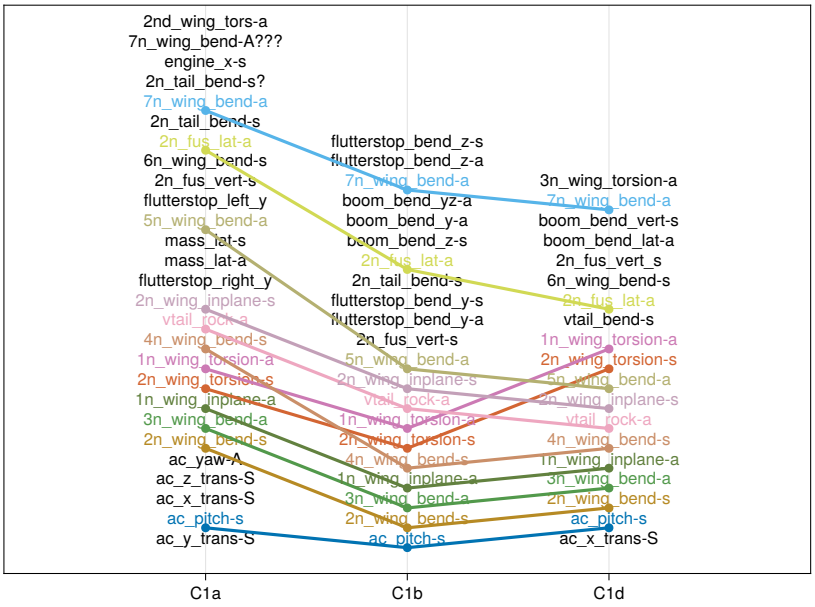
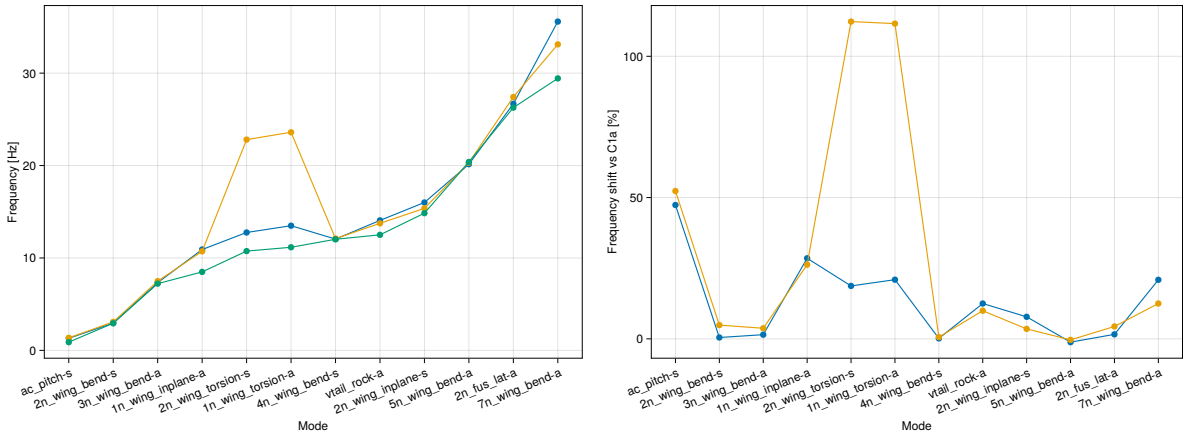


Figure 255: Frequency organization of modal families identified in C1a, C1b and C1d.

some modes identified in C1a are not found in C1b and C1d due to the lower amount of excitations exploited. However, the main observation that can be formulated here is the effect of the flutter stopper assemblies on the torsion modes. In fact, in C1b — flutter stopper masses moved in front position, closer to the torsion line — a clear reordering of modes is obtained, with the first symmetric and antisymmetric torsions moving above the 4 node wing bending mode. In the same conditions, bending modes seem to be very little impacted, as designed, by the mass position modification.

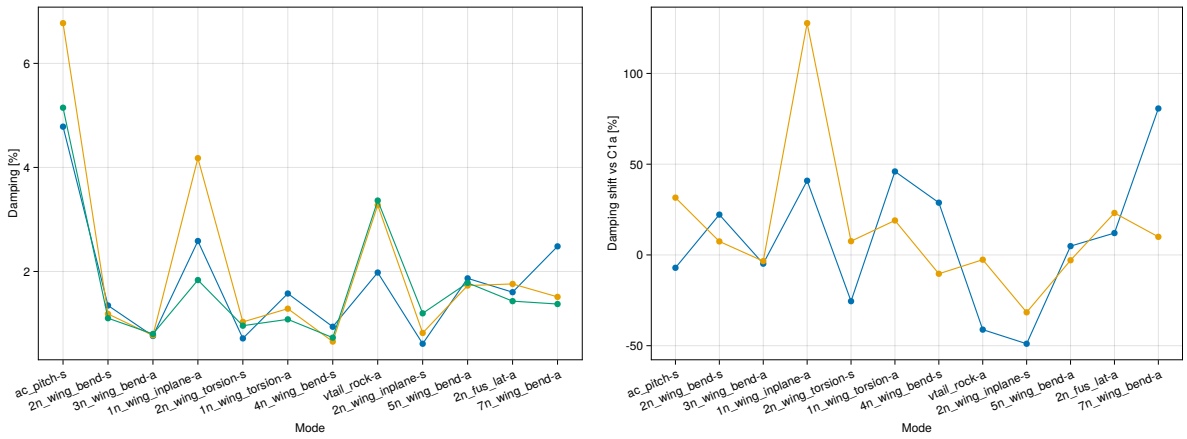
A quantitative comparison of modal frequencies (respectively dampings) is performed in figure 256 (respectively figure 257). A clear trend can be observed in terms of frequencies, where almost only the first torsion modes are influenced by the position, or even the presence, of the flutter stopper devices. Two other modes are influenced by the flutter stopper settings. First, the pitch mode, which is shifted due to the variation in overall mass distribution, which in turns modifies the pitch inertia. Second, the inplane antisymmetric bending mode or so-called scissor mode. This mode is probably affected by the flutter stopper setting because it shows non-planar movement close to a torsion. There is also a much larger effect of the flutter stopper removal than that of the flutter stopper mass position, meaning that the flutter stopper design, might be optimized to reduce the mass of the flutter stopper deviced to obtain a better flutter mass weight/flutter stopper weight. In terms of damping, there is no obivous variation worth commenting.



(a) Absolute values

(b) Relative shift w.r.t C1a.

Figure 256: Comparison of frequency properties of common modes identified in C1a (●●●), C1b (●●●) and C1d (●●●).



(a) Absolute values

(b) Relative shift w.r.t C1a.

Figure 257: Comparison of damping properties of common modes identified in C1a (●●●), C1b (●●●) and C1d (●●●).

GVT - GVT objectives This GVT campagin reached its main objective : characterizing the structural dynamic behavior of the P-FLEX aircraft. An analysis was carried out in the main configuration of interest, the flutter–prone configuration, as well as two secondary confgurations designed to be flutter–free. This campaign allowed to charaterize the efficiency of the flutter stopper design to act on the dyanmic properties of the aircraft. A susbsequent step of this analysis to understand the behavior of the aircraft in flight, is to use the modal identification data from the GVT to perform a flutter simulation. This work is detailed in deliverable [123] of the FLIPASED project.

GVT - Lessons learned While this GVT in itself was not of an extreme complexity, some lessons have been learned in terms of testing both very light and active structures.

Light structures

Testing light structures is relatively complex on its own. In fact, these structures are easily influenced by the surrounding measurement or excitation instrumentation. The use of very light sensors, in order to reduce the intrusivity of the sensor masses lead to the use of non TEDS sensors, which in turn lead to the late detection of instrumentation mis-cabling. It could prove interesting to possess a TEDS capability for very light sensors.

Expectation bias The expected dangerous behavior of the aircraft with the tested wing set, made the test team expect some exceptional phenomena, which mislead us at some points during the test campaing. It should be noted that despite the aspect ratio of this aircraft and its *exotic* features, it still possesses quite a standard behavior. The assumption of exotism of this aircraft lead us to mis-identify an instrumentation problem, which generated slight delays and more important some additional data management tasks.

Time constraint bias Every structural configuration should be excited with several excitation points. Due to time constraints, C1b and C1d were only little explored, leaving some questions on hold. A very minimal and common set of excitations to be performed in each structural configuration should be defined, to make structural configuration comparison more straighforward and comprehensive.

Active control systems

The presence of very active systems on board the aircraft although not changing radically its behavior had numerous negative side effects.

The presence of custom-made high bandwidth actuators and electronics lead to noticeable electro-magnetic compatibility problems. The only feasible solution was then to cut the power to the aileron 4 actuators on both sides of the aircraft. Fortunately the stopping torque and relative low inertia of the ailerons were such that no free motion was observed during the tests.

The presence of high bandwidth actuators also makes the noise analysis more difficult. Indeed, if an actuator is designed to work in the same domain as the one the modal analysis is performed in, some unconventional behavior may be observed. This led to some delays during this camapaign as some of the heavy noise seen on some measurements was initially attributed to the instrumentation setup, while it was actually due to the actuators of the aircraft. In order to assess the presence of noise during the test of such aircraft, a preliminary step of recordings during the power-up of the aircraft systems should be performed systematically, to determine a noise picture and potentially isolated a problematic aircraft system.

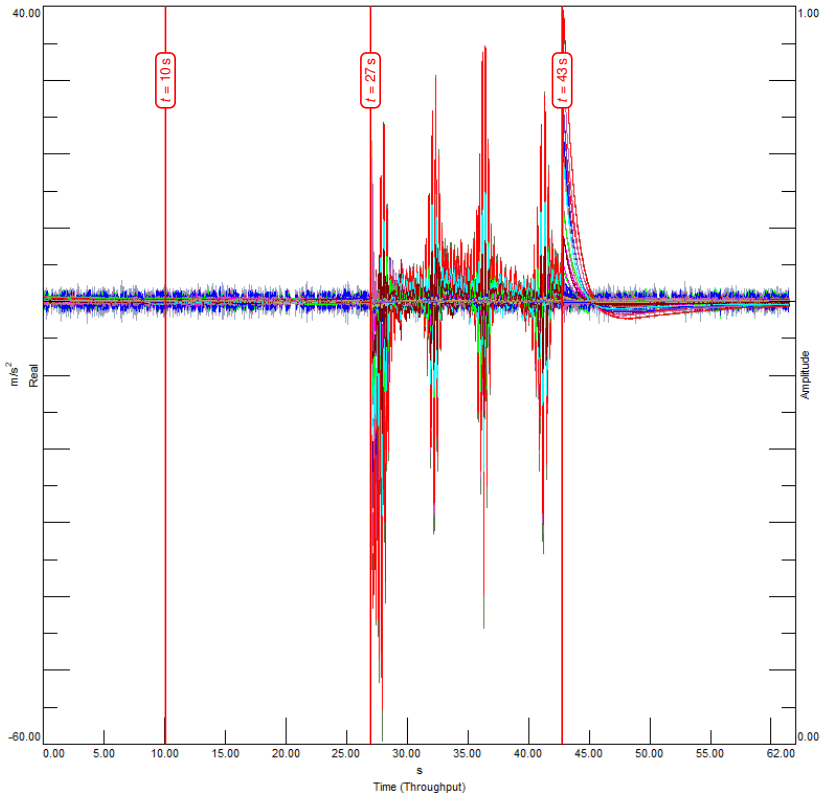
This step was eventually performed on the P-FLEX aircraft at the end of the GVT. Several effects were evaluated : sensor cabling modification, sensor replacement, sensor connection, sensor separation from the structure. None of these effects showed any relevant influence on the large noise level observed. Finally, a full power-up sequence of the Direct Drives was recorded with the following sequence

1. $t = 0$ s Start recording of the GVT channels, as for a standard GVT measurement
2. $t = 10$ s Start power-up sequence of the Direct Drive controllers and actuators
3. $t = 28$ s A noise coming from the Direct Drive actuators was noticeable
4. $t = 43$ s Power-down the Direct Drive controllers and actuactors
5. $t = 62$ s Stop recording

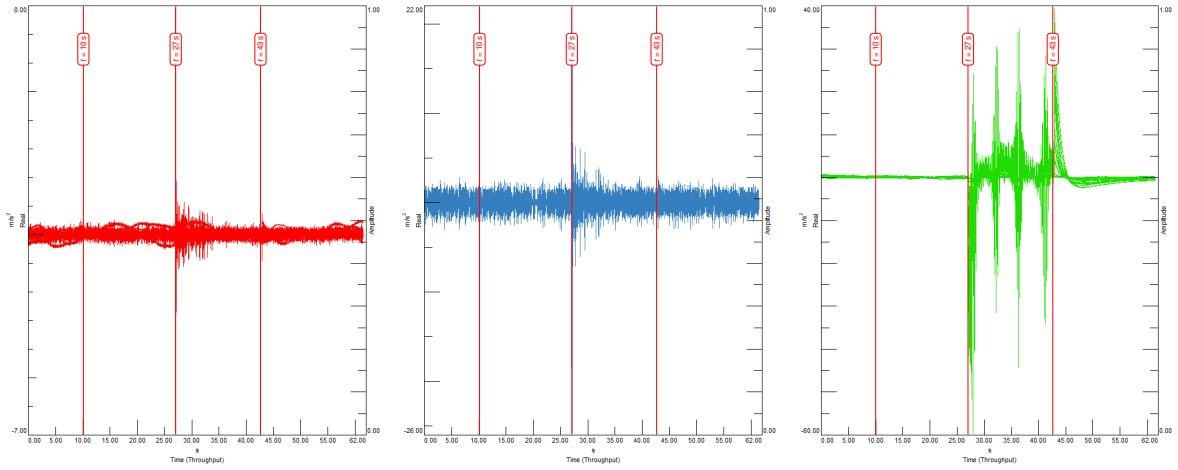
The accelerometer signals recorded during that test are plotted in figure 258. There is an obvious and drastic increase in noise levels, starting from $t = 43$ s coincident with the noise noticed coming from the actuators. This noise can definetly be attributed to electromagnetic pulses emitted by the Direct Drive controller or actuator, as no shaker excitation is performed during this measurement run. As stated above, a measurement was even performed without a sensor connected onto a channel which led to similar observations. Though all sensors showed a specific noise pattern at the power-up of the direct drives, the light triaxis sensors were the most sensitive to this phenomenon, as depicted in figure 258d. On the contrary, the bigger, therefore maybe better shielded Kistler triaxis sensors we less affected, see figure 258b.

Furthermore, testing an aircraft with several and large-bandwidth controllers can lead to a vast increase in test complexity. This is mostly due to the large amount of structure and controller state combinations to test. Also, the aircraft state has to be fully known at each time during the GVT in order to ensure identified data quality. The test planning should account for all of these configurations very precisely. Finally, controller readiness should be considered as critical as airframe readiness and the GVT should be performed only once all controls systems acting on flight or airframe dynamics are in airworthy.

The amount of freeplay and friction in the control surfaces actuation systems led to very complex behavior in the high frequency domain. Though this domain was of little interest for this particular project, it must be recalled that such effects may have very negative consequences on the aircraft lifecycle. In fact, such effect may lead to controller efficiency problems, increased structural fatigue or even unstable dynamic phenomena.



(a) All accelerometers.



(b) Kistler triaxis sensors. **(c)** Light uniaxis sensors. **(d)** Light triaxis sensors.

Figure 258: Time signals of aircraft accelerometers during the power-up test of the Direct Drives. Time key-points (|).

Preparations for flight test campaign As discussed during the winter months, operations of flight testing had to be streamlined. This resulted in going through all of the checklists and trying to find out, which checklist points could be reduced without reducing safety. Checklists for packing and system start-up were improved. Additional equipment like a radio for airport communication, printer and

additional tool kit all helped to decrease the time required to set off for flight testing.

In addition to acquiring new equipment, another pilot has been trained to operate the FLEXOP demonstrator. This required multiple flights with a smaller jet turbine powered glider to be conducted. As a result, three pilots are now available for flight testing, increasing the overall redundancy and flexibility of the flight test crew. Two additional flight test crew members are being trained for Ground Control duties at the moment.

In May 2020 it was decided to start planning the first flight test campaign. There were many unknowns still, due to landing gear not being fully fixed as well as uncertainty due to operational limitations imposed by Covid-19. In June, the first iterations of the landing gear were being tested, but the proper taxi test was only planned for the first day of the flight test campaign. Only the proper high-speed taxi test revealed that the solution applied did not give complete confidence to the pilots. Therefore for the next few days other solutions were being tested on the field, resulting in a need to postpone the actual flight test further to next year. However, the flight planning was already finished for the first campaign and the test cards were fully prepared.

Figure 259: Third pilot preparing for training with a jet turbine sailplane

Flight Permit Acquisition The flight permit application required a new edition of Concept of Operations (CONOPS) written explicitly for the Magdeburg-Cochstedt Airport (EDBC).

The application's first version was submitted to the National Aviation Authority of Germany (LBA) on 31st of August, 2021. The second version had to include some significant changes to the application; the second version was submitted on the 30th of September. After this stage, the feedback regarding the application was only received on the 17th of December and was implemented in the third version

	Flight Test Card	Page II	Flight Test Programme: FLEXOP-FTP-01-00	
	1.9 Baseline Controller Check	Time	CN	Remarks

1.	Engine **ON**	FLEXOP ONE, FLEXOP TWO	*		
2.	REPORT READY FOR TAKE-OFF	MANAGER			
3.	CHECK CONTROLS, FULL DEFLECTIONS	FLEXOP ONE			
4.	JETI WARNINGS **ON**	FLEXOP ONE			
5.	BRAKES **ON**	FLEXOP 1, FLEXOP 2, OPERATOR, ENGINEER			
6.	STANDBY TO ANNOUNCE TAKE-OFF AT 18m/s	OPERATOR			
7.	THROTTLE 100%, BRAKES **OFF** WHEN AIRCRAFT MOVES	FLEXOP 1			T-0
8.	ANNOUNCE **V1**	MANAGER			T+7
9.	FLIGHT STATE CRUISE, THROTTLE 70%, CLIMB 200	FLEXOP 1			At 30 AGL
10.	TRIM 38m/s	FLEXOP 1			
11.	SWITCH AUTOPILOT 1	FLEXOP 1			

Figure 260: First page of the test cards, prepared for the upcoming flights

of the application. The application was submitted on the 13th of January. After receiving the feedback, another version was submitted on 11th of March and the final version on 15th of April. The whole process required significant amount of work on documentation and adjusting the operations that would comply with the new regulations. On top of that, as the requirements were newly introduced during the period of application, it was not clear how to comply with some of the points. For example, the definition of the difference in between the ”sparsely populated area” and ”populated area” is nowhere clearly defined, or the methods to calculate the ground risk buffer was not well described at that point.

Finally, the flight permit was acquired on 25th of April and the detailed planning for the test campaign could take place.

The initial flight permit was issued only from 25th of April till 15th of June. Another update of the permit application had to be done for the permit to be extended from 9th of August till 15th of November. This period was chosen to fit the second and third test campaigns of the year (the third test campaign did not happen due to the accident on flight test 23).

Flight Test Environment (TUM) As Mentioned in the D3.2 - Flight Test Report Phase 1, the rules for flying UAVs in Germany and generally in Europe have changed significantly due to new EASA regulations. This required additional effort to submit an application for a flight permit that would be accepted by the Luftbundesamt (LBA). In 2022 a time-limited flight permit was received after the 5th iteration of the application. Another iteration was required to extend it until the end of the year.

For 2023, 4 more iterations were required. This was due to a change in flight geography, change in the calculation of flight and concingency geographies, some configurational changes and also further information demanded by the LBA. The permit was received until end of May, and then an extension

until end of June.

Overall, the flight permit application process required significant effort from the TUM team, which was not planned in the project. Additionally, it became impossible to receive a permit for Oberpfaffenhofen (EDMO) for this type of aircraft, which meant higher logistical efforts for the flight tests.

Preparations for flying at Magdeburg-Cochstedt Airport (EDBC) The consortium is already familiar with the Magdeburg-Cochstedt Airport (EDBC) airport from last year.

The airport belongs to DLR and is located far away from any major air transport hub, which makes it a good candidate for conducting longer flight test campaigns, even though transportation to Cochstedt would take way longer than to Oberpfaffenhofen.

The airspace details can be seen in Figure 261. The airport layout is seen in Figure 262.

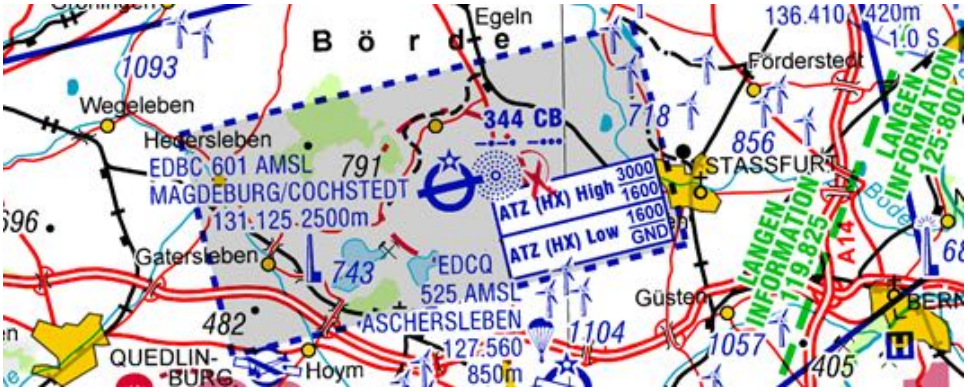


Figure 261: EDBC and the surroundings.

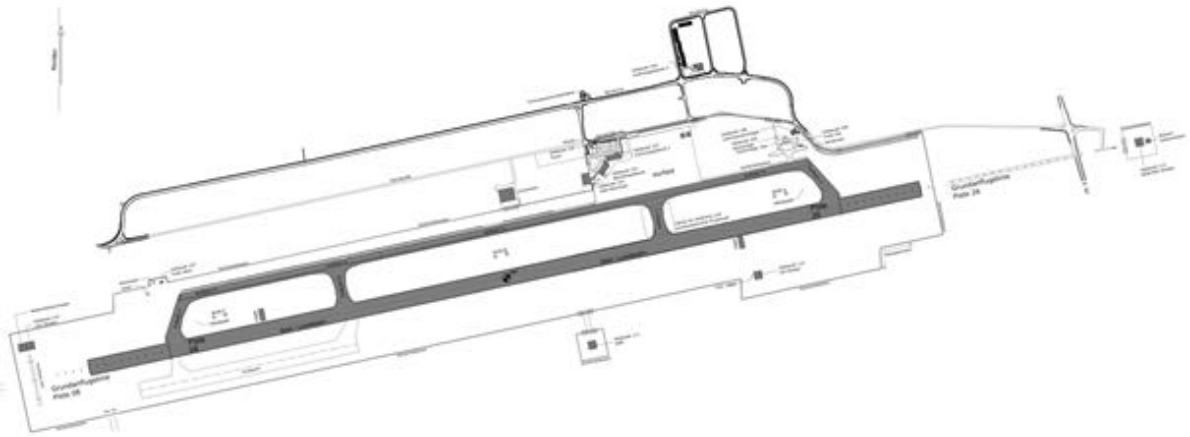


Figure 262: EDBC Airport layout.

EDBC airport has the possibility to close the airspace above it (ATZ (HX) Low and High). T-FLEX flights will only be done when the airspace is closed exclusively for the UAV. A written agreement is received from the EDBC for this. As the airport is at 601ft altitude, the zone has vertical limit of 1000ft (305m) AGL.

The application is made for two circular geographies. The first one is centred around the taxiway Charlie in the east (EDBC I) and the second is centred around the taxiway Delta in the west (EDBC II).

Ground risk buffer calculations for two flight altitudes were calculated (with contingency altitudes in brackets): $H_{FG} = 150m$ ($H_{CV} = 308m$) and $H_{FG} = 300m$ ($H_{CV} = 458m$). To estimate the ground risk buffer, a custom software was used . The software was adjusted with the characteristics of the installed parachute. An example of the single simulation can be found in Figure 263.

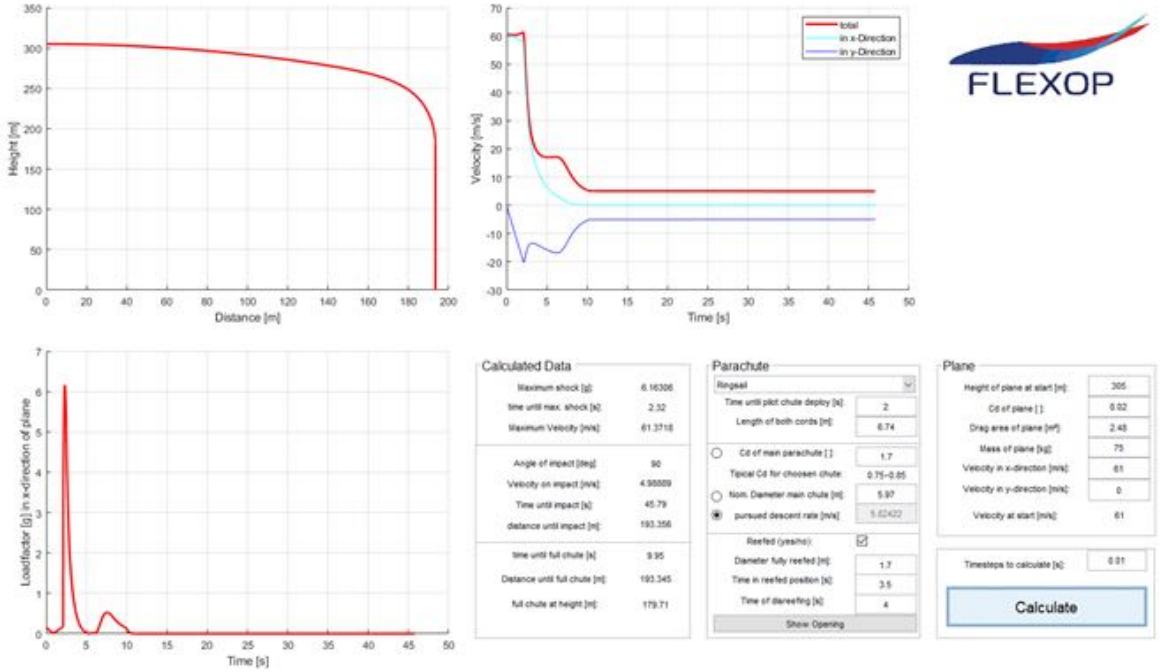


Figure 263: Parachute trajectory when released from 305m AGL, 0m/s wind case.

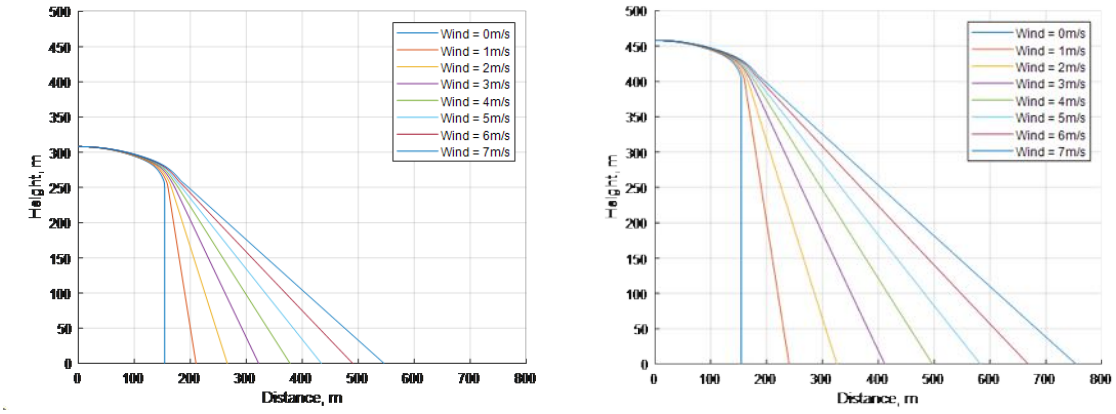
The simulations were performed for various horizontal wind conditions from 0 to 7m/s. The resulting trajectories are presented in Figure 264 and tabulated in Table 43.

Table 43: Distances travelled after parachute release from two different heights

Wind speed, m/s	$H_{CV} = 308m$	$H_{CV} = 458m$
0	155m	155m
1	210m	241m
2	267m	326m
3	323m	412m
4	378m	497m
5	435m	583m
6	491m	669m
7	546m	754m

Due to the big ground risk buffers, different maximum wind speeds had to be selected for the two flight geographies. The first flight geography considers the pilots located at the taxiway Charlie and, taking the surrounding towns into account, allowed for the maximum of 7m/s wind.

The second flight geography considers the pilots located at the taxiway Delta. In this case, due to the proximity of the villages of Neu Königsaue and Cochstedt the available distance in between the pilots



(a) $H_{CV} = 308m$. **(b)** $H_{CV} = 458m$.

Figure 264: Horizontal distance travelled with various wind conditions

and the buildings is reduced to 2040m. Therefore, it was required to also reduce the maximum wind condition to 5m/s (the required ground risk buffer is 583m, Table 43) and also the radius of the flight geography from 1300m to 1260m.

The flight geometries were built in the following manner:

- The pilot position was marked on the map.
- A required flight geography circle was drawn around it (either 1260m or 1300m, which is within the VLOS boundary of 1500m).
- Contingency zone of $S_{CV} = 197m$ was added.
- Maximum operating altitude of $H_{FG} = 300m$ ($H_{CV} = 458m$) was chosen.
- Ground risk buffer zone for the maximum wind condition of either 5m/s or 7m/s at $H_{CV} = 458m$ ($S_{GRB} = 583m$ for 5m/s wind and $S_{GRB} = 754m$ for 7m/s wind) was added.

The resulting maps are shown in Figures 265 and 266.

Flight Test Crew The Flight Test Crew will be made up from members of TUM, as before. Care was taken to have replacement members available for each of the flight crew roles. Due to some of the members leaving TUM permanently, new members were trained.

The training program consists of the following phases:

- Read (and watch) the prepared training material. This includes some background and educational videos based on the previous flight test experience, as well as familiarization with CONOPS and checklists.
- Attend a tour where we go through the aircraft in person and explain all the systems it has.
- Training day in the simulator where we will test the abilities of communication during flight and assign the roles.

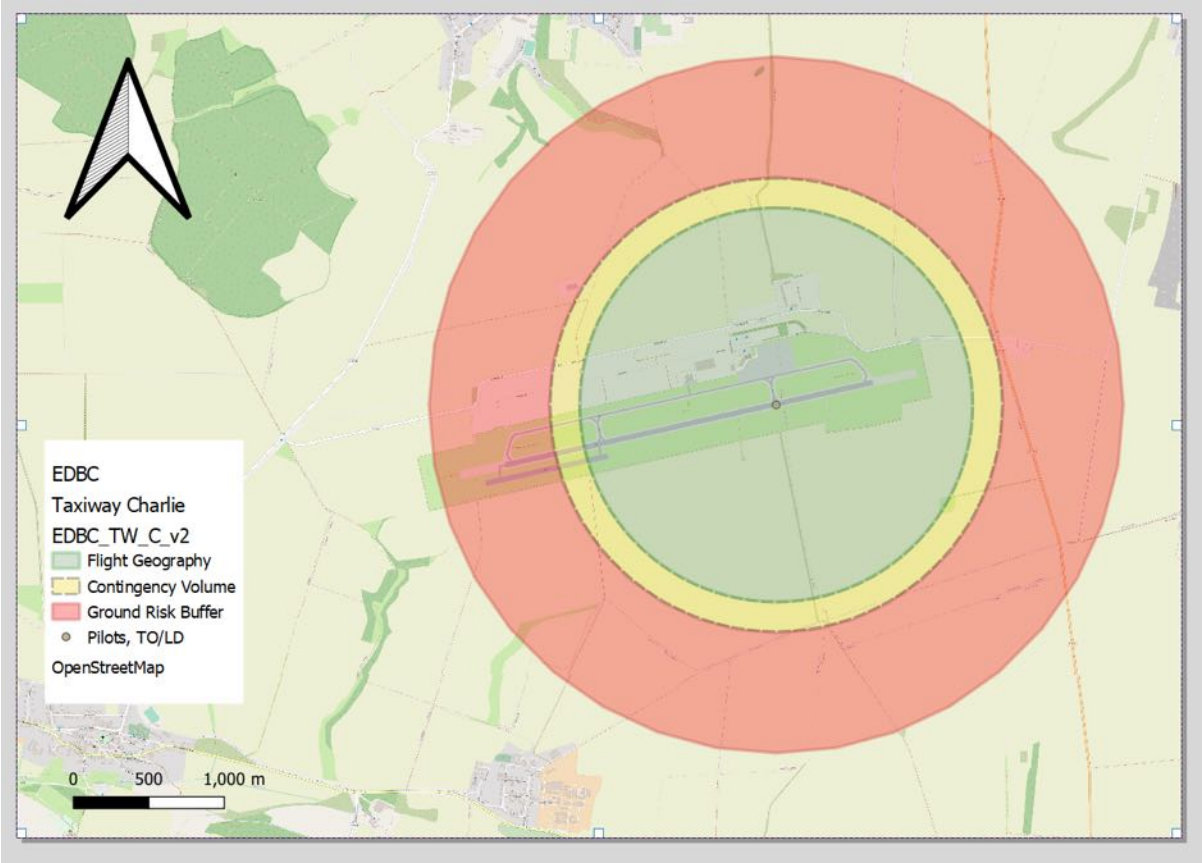


Figure 265: Flight Geography for Taxiway Charlie.

- After that, training for specific roles will be done by the long-time members of the flight test crew.
- We will then do a practise run of the full ground control and aircraft setup, including everything apart from the engine start.
- Time permitting, we might also do a training session for Operator and Manager roles with a real UAV.

Therefore, for the flight season 2023 the flight crew personnel pool consisted of a total of 9 people that were available for different roles. 3 Members could take the role of a pilot, 2 of the flight test manager, 4 of the flight test engineer and 3 of the flight test operator (some members were available for different roles within the team). This provided greater flexibility for planning the campaigns.

Documentation The flight test documentation takes place in three mediums: logged data, video data and written notes.

As before, the flight tests will be logged on the FCC. Additionally, thrust measurement data and operational modal analysis data will be logged on the OBC-II platform. Additionally, the transmitter logs will be added to the main logs. The aircraft configuration parameters will be noted in the acdata.txt file for each flight. Mission Planner log will be available as well.

Videos will be recorded on the two tail-mounted cameras. A 360 degree camera mounted above the

Figure 266: Flight Geography for Taxiway Delta.

fuselage will provide the better overview of the aircraft in-flight. An example of the panoramic view from the camera is shown in Figure 267.

Finally, an external camera tracker will be used, which automatically follows the aircraft during flight.

In 2022 the paper test cards were substituted for their digital versions. The test cards were rewritten in OneNote and are filled out by the Flight Test Manager during flight. This has already proved to be a very convenient change. This allowed to be able to adapt the test cards minutes before the flight, share them with the team online. It also facilitates more efficient pre-flight and post-flight briefing with the team.

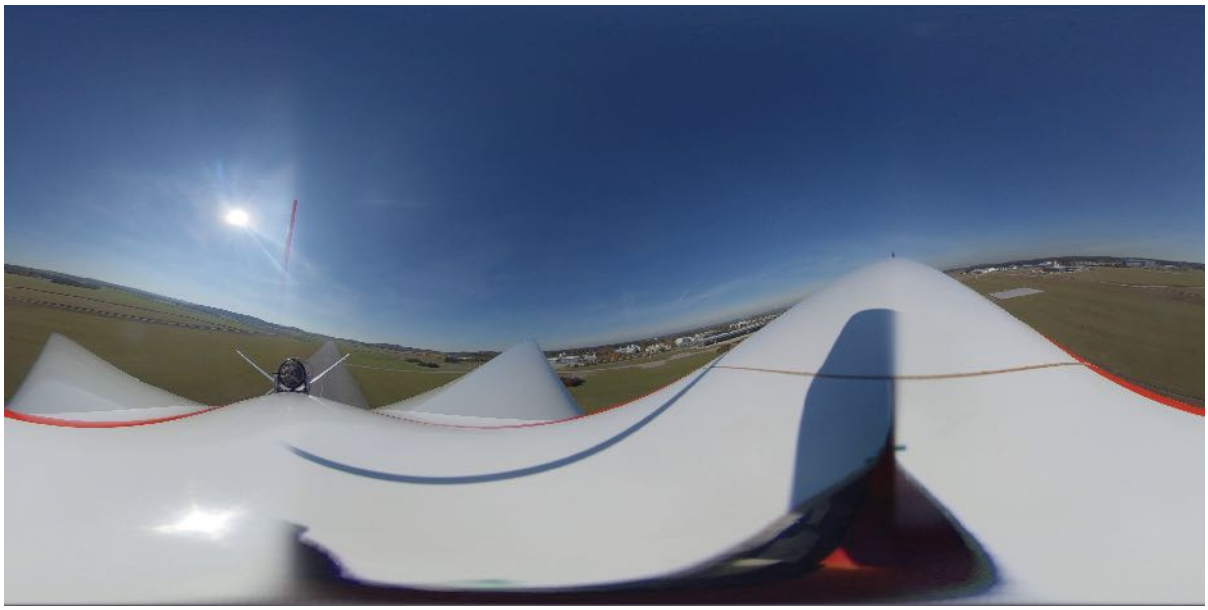


Figure 267: Panoramic view from the 360 degree camera mounted above the fuselage. Visible is (from left to right): left wing, engine and tail, right wing, front-top part of the fuselage.

04 System start-up

Dienstag, 10. Mai 2022 15:56

	System start-up		
ALL	Perform an intercom check	✓	
	Confirm the antenna position is correct.	✓	
OPERATOR	Start the Mission Planner	✓	
OPERATOR	Start the Engineering Data Link	✓	
ENGINEER	Plug in the 6S battery	✓	FLIGHT-HAT and the Raspberry-Pi are booting up and the sensors are getting ready
FLEXOP 1	Start the 360 camera	✓	
ENGINEER	Confirm the flightHAT has a red LED constantly ON	✓	
FLEXOP 1, FLEXOP 2	Confirm that phones are away	✓	
FLEXOP 1, FLEXOP 2	Both transmitters ON	✓	
FLEXOP 1, FLEXOP 2	Transmitter model FLEXOP 1	✓	

18 02

Figure 268: An example of the test cards used.

Goals for the 1st flight test campaign Due to the problems described in D3.2 Flight Test Report – Flight Test Phase #1, only three test flights were conducted in years 2020 and 2021 (designated FT7, FT8 and FT9). Therefore the first flight of the test campaign in 2022 would be the fourth flight of the project (designated FT10).

The following goals (in decreasing priority) were raised for the campaign:

1. Test all the functionalities of the autopilot so that it is ready for flutter tests. This means testing the following modules:
   (a) Augmented mode,
   (b) Altitude hold,
   (c) Airspeed hold,
   (d) Course angle hold,
   (e) Coordinated turn,
   (f) Waypoint flight (following the race track),
   (g) Signal injection.
2. Gather enough flight data to identify the rigid body modes. Special manoeuvers (pulses, doublets, multisine inputs, pull-ups) were planned for this.
3. Gather enough flight data to identify the flexible body modes. No special manoeuvers were planned for this.
4. Baseline aerodynamic coefficient determination:
   (a) Lift/drag/moment polars for cruise configuration (part of the data from point 2 can be used),
   (b) Gather in-flight data for take-off and landing flap configurations,
   (c) Gather data for drag components (landing gear and different airbrake settings).
5. Flying the -1 wing. Flying it in a flutter-safe configuration would give us confidence for the future flights with it. If it is flown, manoeuvers for rigid body modes, baseline aerodynamics and flexible modes would be done.
6. Training new pilots. Two new pilots have joined the test team at TUM who have never flown the T-FLEX before. With one of the old pilots leaving TUM soon, the new pilots must get confident with the aircraft.
7. Visualization of airflow over the wings:
   (a) Tuft experiments are planned and prepared to look for any unexpected flow separation on -0 wing,
   (b) Oil flow experiments are planned and prepared to investigate transition location on -0 wing. No extra manoeuvers were planned for this.
8. Airspeed calibration in-flight. Manoeuvers to identify the position error of the air data system are prepared.
9. Induced drag experiments. Performing flights with -0 wing with different flap configurations could already give insight if the change of induced drag is measurable. For this reason, three flap configurations would be flown: standard cruise state, maximum induced drag state and minimum induced drag state.
10. Engine model identification. Further throttle injections can be done, if required.

Goals for the 2nd flight test campaign After completing the 1st flight test campaign in May, planning was done for the next campaign in August. There were changes planned with the main flight control computer of the aircraft (a new RX-MUX-II hardware unit was being assembled at that time), autothrottle controller had to be retuned. It was also planned to test a new, less complex procedure for the upcoming flutter tests. The new procedure suggested that instead of flying horse-race patterns with a short measurement leg (which was a maximum of 15s), a steady coordinated turn at a low bank angle should be flown. Without having to turn, accelerate and decelerate, the measurement leg could be made longer, which was required for online modal analysis method to work properly.

The reviewed goals for the second campaign were as follows:

1. The baseline -0 wing:
   (a) Test the new RXMUXv2 hardware in-flight and confirm that everything is working as before.
   (b) Tune the autothrottle controller.
   (c) Do additional flights for rigid mode identification with injected manoeuvers on -0 wing.
   (d) Test the new suggested flutter procedure (big 800+m radius circles with increasing airspeed).
   (e) Test the new altitude hold mode.
2. The flexible -1 wing:
   (a) Maiden flight with the -1 wing.
   (b) Autopilot tuning for the -1 wing (not expected to differ much).
   (c) Rigid and flexible mode identification with -1, aerodynamic model update.
   (d) Direct drive (flutter suppression actuator) tests in-flight.
3. Additional goals:
   (a) Pilot training.
   (b) Investigate landing gear drag.
   (c) Investigate the FLAP1 separation effects on the V-Tail.
   (d) Investigate the engine wake effects on the V-Tail.
   (e) Effect of FLAP2 and FLAP3 on the pitching moment.

Goals for the 3rd Flight Test Campaign The goals for the 3rd Flight Test Campaign were split into primary for the flight dynamics, primary for the autopilot system, and secondary objectives.

The primary flight dynamics goals for the first flight test campaign are (responsible partners are noted down in the brackets):

- Perform the maiden flight with the new demonstrator. (TUM, SZTAKI)
- Confirm take-off and landing performance for the new demonstrator. (TUM)
- Confirm fuel system performance (main and auxiliary), and evaluate the increase in flight time and c.g. shift due to fuel transfer (TUM)
- Confirm the sensor network works as required. (SZTAKI)
- Get new trim values for all flight states (take-off, cruise, landing). (TUM)

- Familiarize with handling qualities of the new demonstrator. (TUM)
- Direct drive modelling in-flight (frequency sweeps at different loads/airspeeds). (SZTAKI)
- Perform manoeuvers for baseline performance evaluation (lift, drag polars). (TUM)
- Perform manoeuvers for rigid and elastic model identification and update. (TUM, DLR-SR)
- Test the flutter stopper function in-flight (for this we would need to fly flutter configuration, not flutter-free). (TUM, DLR-SR, DLR-G, SZTAKI)
- Test the data processing and evaluation toolchain in-flight (telemetry links, online data retrieval, model updating). (TUM, DLR-SR, DLR-G, SZTAKI)

The primary autopilot system goals are:

- Tune the augmented mode for the new demonstrator. (SZTAKI)
- Confirm that the altitude hold mode can track the required pressure altitude. (SZTAKI)
- Adjust the flight control surface lookup tables and confirm their behavior (SZTAKI, TUM)
- Confirm that the autothrottle mode can follow different airspeeds all the way up to the second flutter speed. (SZTAKI)
- Confirm that the autopilot can guide the aircraft in circles (getting into the circle, following the circle, aborting the circle). (SZTAKI)
- Confirm that all three modes can work together (increase the airspeed while flying in a circle). (SZTAKI)
- Test and refine the telemetry and Engineering Data Link (EDL) system (SZTAKI, TUM)

Finally, some secondary goals were noted down:

- Investigate the engine wake effects on the V-Tail. (TUM)
- Investigate the effects of asymmetry of the wings and the possible associated free-play (TUM, DLR-G).

Goals for the 4th Flight Test Campaign The goals of the 4th and final Flight Test Campaign of the project FLIPASED followed the outcomes of the 3rd campaign. Many goals connected to system checks of the demonstrator were already achieved.

The primary flight dynamics goals for this campaign were (responsible partners are noted down in the brackets):

- Direct drive modelling in-flight (frequency sweeps at different loads/airspeeds). (SZTAKI)
- Perform manoeuvers for baseline performance evaluation (lift, drag polars). (TUM)
- Perform manoeuvers for rigid and elastic model identification and update. (TUM, DLR-SR)
- Test the flutter stopper function in-flight (for this we would need to fly flutter configuration, not flutter-safe). (TUM, DLR-SR, DLR-G, SZTAKI)

- Test the data processing and evaluation toolchain in-flight (telemetry links, online data retrieval, model updating). (TUM, DLR-SR, DLR-G, SZTAKI)

The primary autopilot system goals were:

- Confirm that the autothrottle mode can follow different airspeeds all the way up to the second flutter speed. (SZTAKI)
- Confirm that the autopilot can guide the aircraft in circles (getting into the circle, following the circle, aborting the circle). (SZTAKI)
- Confirm that all three modes can work together (increase the airspeed while flying in a circle while holding the altitude). (SZTAKI)

The primary flutter control goals were:

- Confirm the DLR flutter controller increases wing damping in the subcritical region (under 42m/s). (DLR, SZTAKI)
- Confirm the SZTAKI flutter controller increases wing damping in the subcritical region (under 42m/s). (SZTAKI)
- Confirm the DLR flutter controller increases wing damping in flutter region (above 50m/s). (DLR)
- Confirm the SZTAKI flutter controller increases wing damping in flutter region (above 50m/s). (SZTAKI)
- Validate open-loop flutter speed prediction of the models. (ONERA, DLR)

Finally, some secondary goals were noted down which were not investigated during the previous campaign:

- Investigate the engine wake effects on the V-Tail. (TUM)
- Investigate the effects of asymmetry of the wings and the possible associated freeplay. (DLR, ONERA)
- Induced drag reduction with active load shape control. (DLR, SZTAKI)

Planning for the 1st flight test campaign The goals mentioned above were roughly allocated to specific flight tests for initial planning. Some test points were already assigned to specific flights: 3 Autopilot test flights were estimated (Figure 269), 4 baseline aerodynamic flights (two of them are described in Figure 270) and 6 rigid body mode aerodynamic identification flights (Figures 271 and 272).

In the end, the test points were discussed before each flight and test plan had to be rewritten due to many unforeseen problems.

Initially it was estimated that at the best case scenario 4 flights could be done per day. Combined with the availability of the crew, a schedule was made (Figure 273). In the schedule the number of flights done is also noted.

	A	B	C	D	E	F	G	H	I	J
1	ID	AP1			AP2			AP3		
2	Goal	Augmented mode check, pilot training			Autothrottle check			Course angle and horse race pattern		
3	Number of test points	9			10			10		
4	Planned duration	00:19:00			00:15:00			00:14:30		
5	1	Take-off		00:00:30	Take-off		00:00:30	Take-off		00:00:30
6	2	Free flight	Pilot training	00:05:00	Trim point	38m/s	00:00:30	Trim point	38m/s	00:00:30
7	3	Controller check	augmented mode, old LUT	00:01:00	Controller check	Robust, 38	00:01:00	Controller check	Augmented, autothrottle, alt hold	00:01:00
8	4	Controller check	augmented mode, new LUT	00:01:00	Controller check	Robust, 38 > 42	00:01:00	Controller check	East/West	00:01:00
9	5	Controller check	Altitude hold	00:01:00	Controller check	Performance, 38	00:01:00	Controller check	Coordinated turn	00:01:00
10	6	Controller check	Altitude hold + 25m	00:01:00	Controller check	Performance, 38 > 42	00:01:00	Trim point	38m/s	00:00:30
11	7	Controller check	Altitude hold - 25m	00:01:00	Controller check	Performance, 42, + 25m	00:01:00	Controller check	Horse race, 38 CW	00:01:00
12	8	Landing imitation		00:02:00	Controller check	TECS, 38 > 42	00:01:00	Controller check	Horse race, 42 CW	00:01:00
13	9	Landing		00:02:00	Controller check	TECS, 42, + 25m	00:01:00	Controller check	Horse race, 38 CCW	00:01:00
14	10				Landing		00:02:00	Landing		00:02:00
15	11									
16	12									
17	13									
18	14									
19	15									
20	16									
21	17									
22	18									
23	19									
24	20									

Figure 269: Test points planned for autopilot controller tests.

	A	B	C	D	E	F	G
1	ID	AE1			AE2		
2	Goal	Steady state test points - TO and Landing			Airspeed calibration		
3	Number of test points	11			8		
4	Planned duration	00:15:30			00:18:30		
5	1	Take-off		00:00:30	Take-off		00:00:30
6	2	Trim point	30m/s, Take-off	00:00:30	Trim point	30m/s	00:00:30
7	3	Acceleration-deceleration	Trim point with take-off configuration	00:01:00	Airspeed calibration - JMOSS	Accelerate from 30 to 60	00:02:30
8	4	Acceleration-deceleration	Trim point with take-off configuration	00:01:00	Airspeed calibration - JMOSS	Accelerate from 30 to 60	00:02:30
9	5	Trim point	30m/s, Landing	00:00:30	Airspeed calibration - JMOSS	Accelerate from 30 to 60	00:02:30
10	6	Acceleration-deceleration	Trim point with landing configuration	00:01:00	Airspeed calibration - cloverleaf	40	00:02:00
11	7	Acceleration-deceleration	Trim point with landing configuration	00:01:00	Airspeed calibration - cloverleaf	50	00:02:00
12	8	Trim point	30m/s, Cruise	00:00:30	Landing		00:02:00
13	9	Acceleration-deceleration	Trim point with cruise configuration	00:01:00			
14	10	Acceleration-deceleration	Trim point with cruise configuration	00:01:00			
15	11	Landing		00:02:00			
16	12						
17	13						
18	14						
19	15						
20	16						
21	17						
22	18						
23	19						
24	20						

Figure 270: Test points planned for baseline aerodynamic tests.

	A	B	C	D	E	F	G	H	I	J
1	ID	RID1			RID2			RID3		
2	Goal	Rigid mode dynamics identification 1			Rigid mode dynamics identification 2			Rigid mode dynamics identification 3		
3	Number of test points	16			16			16		
4	Planned duration	00:17:30			00:17:30			00:17:30		
5	1	Take-off		00:00:30	Take-off		00:00:30	Take-off		00:00:30
6	2	Elevator pulse	42	00:00:30	Elevator pulse	38	00:00:30	Elevator pulse	34	00:00:30
7	3	Elevator pulse	42	00:00:30	Elevator pulse	38	00:00:30	Elevator pulse	34	00:00:30
8	4	Elevator pulse	42	00:00:30	Elevator pulse	38	00:00:30	Elevator pulse	34	00:00:30
9	5	Elevator doublet	42	00:00:30	Elevator doublet	38	00:00:30	Elevator doublet	34	00:00:30
10	6	Elevator doublet	42	00:00:30	Elevator doublet	38	00:00:30	Elevator doublet	34	00:00:30
11	7	Elevator doublet	42	00:00:30	Elevator doublet	38	00:00:30	Elevator doublet	34	00:00:30
12	8	Multisine input (elevator)	42	00:00:30	Multisine input (elevator)	38	00:00:30	Multisine input (elevator)	34	00:00:30
13	9	Multisine input (elevator)	42	00:00:30	Multisine input (elevator)	38	00:00:30	Multisine input (elevator)	34	00:00:30
14	10	Multisine input (elevator)	42	00:00:30	Multisine input (elevator)	38	00:00:30	Multisine input (elevator)	34	00:00:30
15	11	Pushover-pull-up	42	00:00:30	Pushover-pull-up	38	00:00:30	Pushover-pull-up	34	00:00:30
16	12	Pushover-pull-up	42	00:00:30	Pushover-pull-up	38	00:00:30	Pushover-pull-up	34	00:00:30
17	13	Pushover-pull-up	42	00:00:30	Pushover-pull-up	38	00:00:30	Pushover-pull-up	34	00:00:30
18	14	Pushover-pull-up	42	00:00:30	Pushover-pull-up	38	00:00:30	Pushover-pull-up	34	00:00:30
19	15	Pushover-pull-up	42	00:00:30	Pushover-pull-up	38	00:00:30	Pushover-pull-up	34	00:00:30
20	16	Landing		00:02:00	Landing		00:02:00	Landing		00:02:00
21	17									
22	18									
23	19									
24	20									

Figure 271: Test points planned for rigid mode identification tests (flights 1 to 3).

Planning for the 2nd flight test campaign Following the flight campaign goals, the following flight plan was suggested:

	A	K	L	M	N	O	P	Q	R	S
1	ID	RID4			RID5			RID6		
2	Goal	Rigid mode dynamics identification 4			Rigid mode dynamics identification 5			Rigid mode dynamics identification 6		
3	Number of test points	16			16			16		
4	Planned duration	00:17:30			00:17:30			00:17:30		
5	1	Take-off		00:00:30	Take-off		00:00:30	Take-off		00:00:30
6	2	Rudder doublet	34	00:00:30	Rudder doublet	38	00:00:30	Rudder doublet	42	00:00:30
7	3	Rudder doublet	34	00:00:30	Rudder doublet	38	00:00:30	Rudder doublet	42	00:00:30
8	4	Rudder doublet	34	00:00:30	Rudder doublet	38	00:00:30	Rudder doublet	42	00:00:30
9	5	Aileron doublet	34	00:00:30	Aileron doublet	38	00:00:30	Aileron doublet	42	00:00:30
10	6	Aileron doublet	34	00:00:30	Aileron doublet	38	00:00:30	Aileron doublet	42	00:00:30
11	7	Aileron doublet	34	00:00:30	Aileron doublet	38	00:00:30	Aileron doublet	42	00:00:30
12	8	Multisine input (ailerons)	34	00:00:30	Multisine input (ailerons)	38	00:00:30	Multisine input (ailerons)	42	00:00:30
13	9	Multisine input (ailerons)	34	00:00:30	Multisine input (ailerons)	38	00:00:30	Multisine input (ailerons)	42	00:00:30
14	10	Multisine input (ailerons)	34	00:00:30	Multisine input (ailerons)	38	00:00:30	Multisine input (ailerons)	42	00:00:30
15	11	Multisine input (rudder)	34	00:00:30	Multisine input (rudder)	38	00:00:30	Multisine input (rudder)	42	00:00:30
16	12	Multisine input (rudder)	34	00:00:30	Multisine input (rudder)	38	00:00:30	Multisine input (rudder)	42	00:00:30
17	13	Multisine input (rudder)	34	00:00:30	Multisine input (rudder)	38	00:00:30	Multisine input (rudder)	42	00:00:30
18	14	Sideslip	34	00:00:30	Sideslip	38	00:00:30	Sideslip	42	00:00:30
19	15	Sideslip	34	00:00:30	Sideslip	38	00:00:30	Sideslip	42	00:00:30
20	16	Landing		00:02:00	Landing		00:02:00	Landing		00:02:00
21	17									
22	18									
23	19									
24	20									

Figure 272: Test points planned for rigid mode identification tests (flights 4 to 6).

Date	Day	Max flight slots	Actual flights	Flight Slot 1	Flight Slot 2	Flight Slot 3	Flight Slot 4
09.05.2022	Monday	2	1	AP1	AP2		
10.05.2022	Tuesday	4	0	RID1	RID2	RID3	
11.05.2022	Wednesday	4	0	AP3	RID4	RID5	
12.05.2022	Thursday	3	0	RID6	AE1		
13.05.2022	Friday	4	0	AE2			
14.05.2022	Saturday	4	0				
15.05.2022	Sunday	3	0				
16.05.2022	Monday	4	2				
17.05.2022	Tuesday	4	2				
18.05.2022	Wednesday	4	1				
19.05.2022	Thursday	3	2				

Figure 273: The planned schedule for the flight test campaign.

1. FT18: Pilot training + pushover-pullup training
2. FT19: RXMUX2 testing + Autothrottle tuning
3. FT20: Rigid body mode manoeuver injections (calm weather)
4. FT21: Rigid body mode manoeuver injections (calm weather)
5. FT22: Flutter testing procedure + Envelope expansion (banked turn up to 60m/s)
6. FT23: -1 Wing maiden flight + system check
7. FT24: -1 Wing autopilot mode testing
8. FT25: -1 Wing system identification manoeuvers

Planning for the 3rd Flight Test Campaign The detailed planning of the campaign started on 28.02. However, at that point it was realized that keeping the original schedule (to perform the 3rd campaign on 20-31.03) was impossible due to the following reasons:

- The taxi tests have not yet been performed. This was planned (weather permitted) for the week after the GVTs.
- At that point there were still no updates about the flight permit form the LBA.

- Due to the slip in GVT schedule (originally the GVT was planned for 27.02-10.03, in reality it took place 06-17.03), there was only one week in between the GVTs and the flight tests. It would not be enough time for the flight test crew to prepare the aircraft.

Consequently, the flight test campaign was shifted back by one week. This would make the campaign one day shorter, because the Friday on the second week is a holiday. But would give more time to make sure everything works before actually going there. The following plan was made:

- 26.03 – The aircraft gets transported to Cochstedt.
- 27.03 – The aircraft is assembled and we can do full system tests that are necessary. Maybe a taxi test.
- 28.03 – Flight tests start.
- 01.04 – A day off. Only in a worse case scenario it was planned to perform tests on Saturday.
- 02.04 – A day off.
- 03.04 – Testing continues.
- 05.04 – Last flight tests are done (7 flight days in total). Packing is done.
- 06.04 – The aircraft is transported back to Munich.

Before the campaign, three flight tests were planned in detail – the maiden flight, the baseline controller check flight and aerodynamic system identification flight. These flights were already described in the Deliverable D3.8[5].

Planning for the 4th Flight Test Campaign To allow more time for data processing and analysis during the second campaign, a one-week break was planned between the first and second flight week of the second campaign. It was also taken into account that if anything goes wrong during the first week, the second week would be used for additional tests or repairs. This meant that the first week of the second campaign would take place on 08-12.05 and the second week (data analysis/additional tests/repair week) would be on 15-19.05. The final flight test week of the project would take place on 22-26.05.

The schedule of the three weeks was:

- 07.05 – Arrive to EDBC. The teams should be at EDBC around 14-15:00, so that some of the preparation tasks can be done before Monday.
- 08.05 – Our flight window opens.
- 12.05 – Flight can continue until around 18:00.
- (15-17.05 – Second flight week. These are reserve days. If something goes really bad, we might have to go to EDBC to either fix things or do flights. This will be discussed in the course of the first week).
- 21.05 – Arrive to EDBC again.
- 22.05 – Third flight week opens.
- 26.05 – Last flight day for the project. Packing follows.

For planning purposes the list of manoeuvers to be injected (or flown manually) was prepared and split into different blocks (Table 44). The goals mentioned in previous section were further divided among the flights:

Scheduled flights (1st week):

- FT26 (flutter safe configuration). Autothrottle tests, coordinated turn. Going through the manoeuver list (block 1).
- FT27 (flutter safe configuration). Flutter controller test point (check for operation, reception, visibility). Going through the manoeuver list (block 2).
- FT28 (flutter safe configuration). Going through the manoeuver list (block 3).
- FT29 (flutter configuration). Flutter controller tests in subcritical region. Selected manoeuvers at low speed (block 4). Flutter stopper test.

Scheduled flights (3rd week):

- FT30 (flutter configuration). Flutter controller tests in subcritical region.
- FT31 (flutter configuration). Flutter controller tests in critical region.
- FT32 (flutter configuration). Flutter controller tests in critical region.
- FT33 (flutter configuration). Flutter controller tests in critical region.

Table 44: Manoeuvers planned for the final flight test campaign, distributed in different blocks.

Nr	Block	Title	Amplitude	Airspeed, m/s	Targets
		Manual manoeuvers			
1	1	Pushover-pullup		36	0-2G, 10deg/s, 20deg pitch
2	1	Pushover-pullup		36	0-2G, 10deg/s, 20deg pitch
3	1	Pushover-pullup		36	0-2G, 10deg/s, 20deg pitch
4	2	Acceleration-deceleration		30-50-30	
5	2	Acceleration-deceleration		30-50-30	
6	2	Acceleration-deceleration		30-50-30	
7	3	Pushover-pullup		46	0-2G, 10deg/s, 20deg pitch
8	3	Pushover-pullup		46	0-2G, 10deg/s, 20deg pitch
9	3	Pushover-pullup		46	0-2G, 10deg/s, 20deg pitch
10	3	Banked manual doublet		36	+40deg bank, 10deg/s
11	3	Banked manual doublet		36	+40deg bank, 10deg/s
12	3	Banked manual doublet		36	-40deg bank, 10deg/s
13	3	Banked manual doublet		36	-40deg bank, 10deg/s
		Automated injections			
14	1	Multisine elevator	5	36	
15	1	Multisine elevator	5	36	
16	1	Flap 2-3-4 multisine	3	36	
17	1	Flap 2-3-4 multisine	3	36	
18	1	Multisine rudder	5	36	
19	1	Multisine rudder	5	36	
20	1	Multisine aileron	3	36	
21	1	Multisine aileron	3	36	
22	1	Asymmetric chirp	4	36	
23	1	Asymmetric chirp	4	36	
24	1	Symmetric chirp	4	36	
25	1	Symmetric chirp	4	36	
26	2	Multisine elevator	4	46	
27	2	Multisine elevator	4	46	
28	2	Flap 2-3-4 multisine	3	46	
29	2	Flap 2-3-4 multisine	3	46	
30	2	Multisine rudder	4	46	
31	2	Multisine rudder	4	46	
32	2	Multisine aileron	3	46	
33	2	Multisine aileron	3	46	
34	2	Symmetric Aileron doublet 1	5	36	
35	2	Symmetric Aileron doublet 1	5	36	
36	2	Asymmetric Aileron doublet 1	5	36	
37	2	Asymmetric Aileron doublet 1	5	36	
38	2	Symmetric Aileron doublet 2	5	36	
39	2	Symmetric Aileron doublet 2	5	36	
40	2	Asymmetric Aileron doublet 2	5	36	
41	2	Asymmetric Aileron doublet 2	5	36	
42	3	Symmetric chirp	3	46	
43	3	Symmetric chirp	3	46	
44	3	Asymmetric chirp	3	46	
45	3	Asymmetric chirp	3	46	
46	4	Asymmetric chirp	4	36	
47	4	Asymmetric chirp	4	36	
48	4	Symmetric chirp	4	36	
49	4	Symmetric chirp	4	36	

Flight Testing After numerous iterations of the landing gear the pilots agreed that the aircraft is controllable. This allowed to continue the flight tests.

The first flight test under the name of the project was therefore conducted on 21st of April, 2021. Flight lasted for around 10 minutes and the autothrottle functionality was tested in the air. The graphs of the flight test can be seen below:

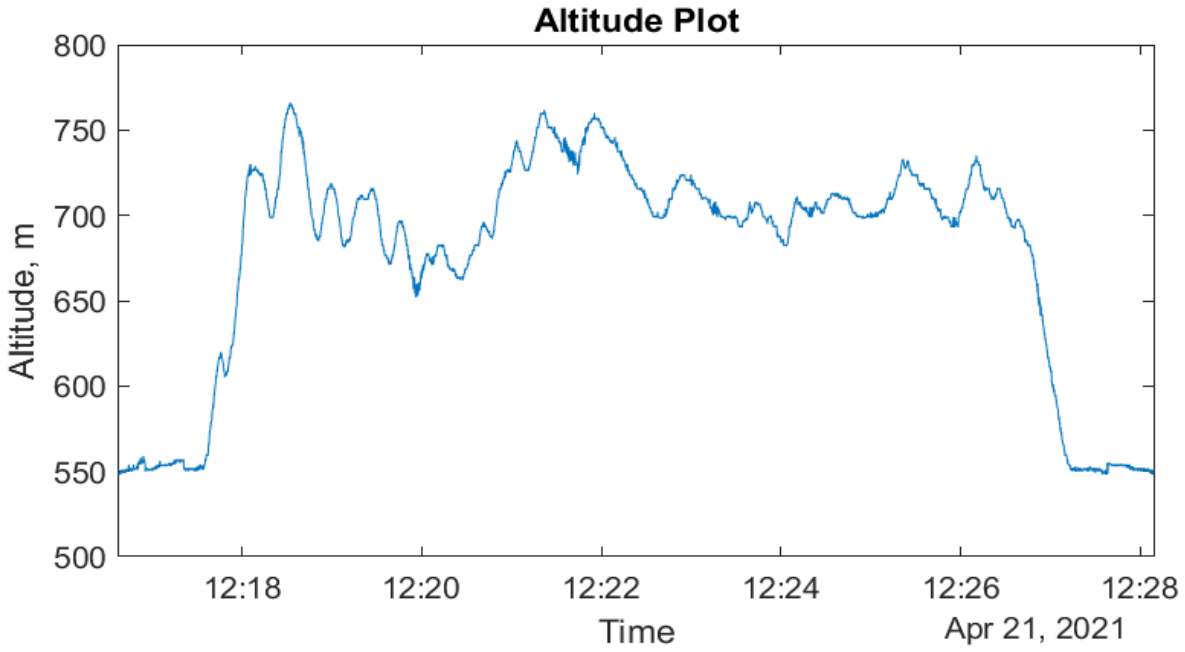


Figure 274: Flight test data

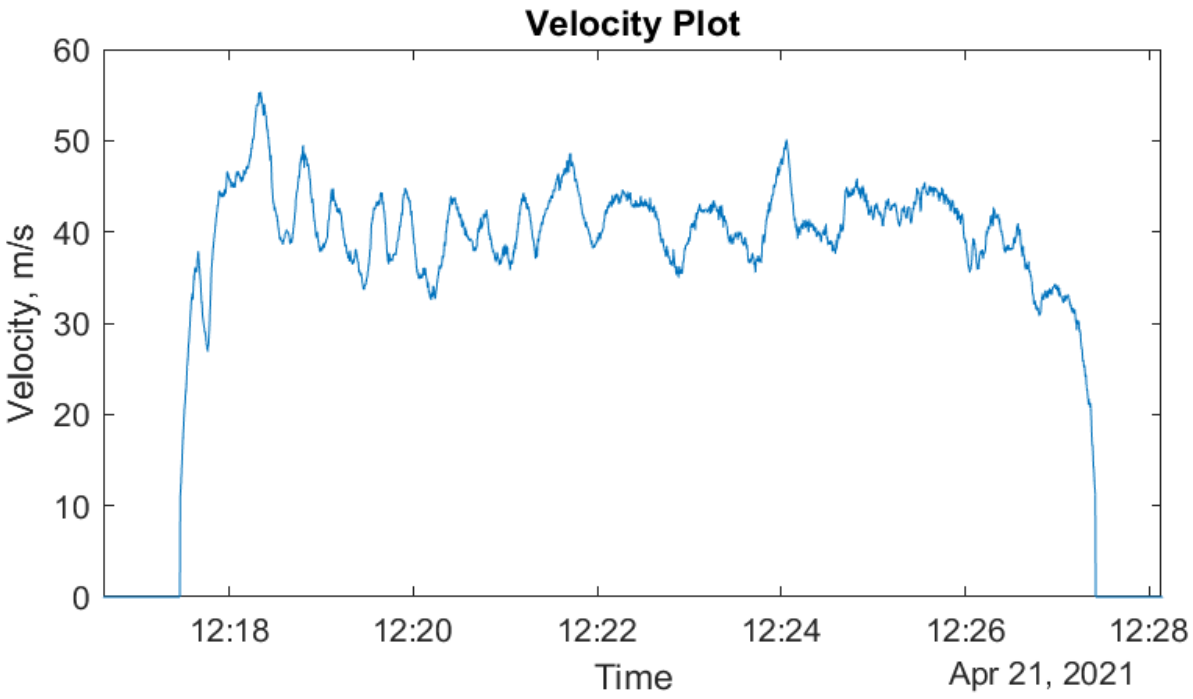


Figure 275: Flight test data

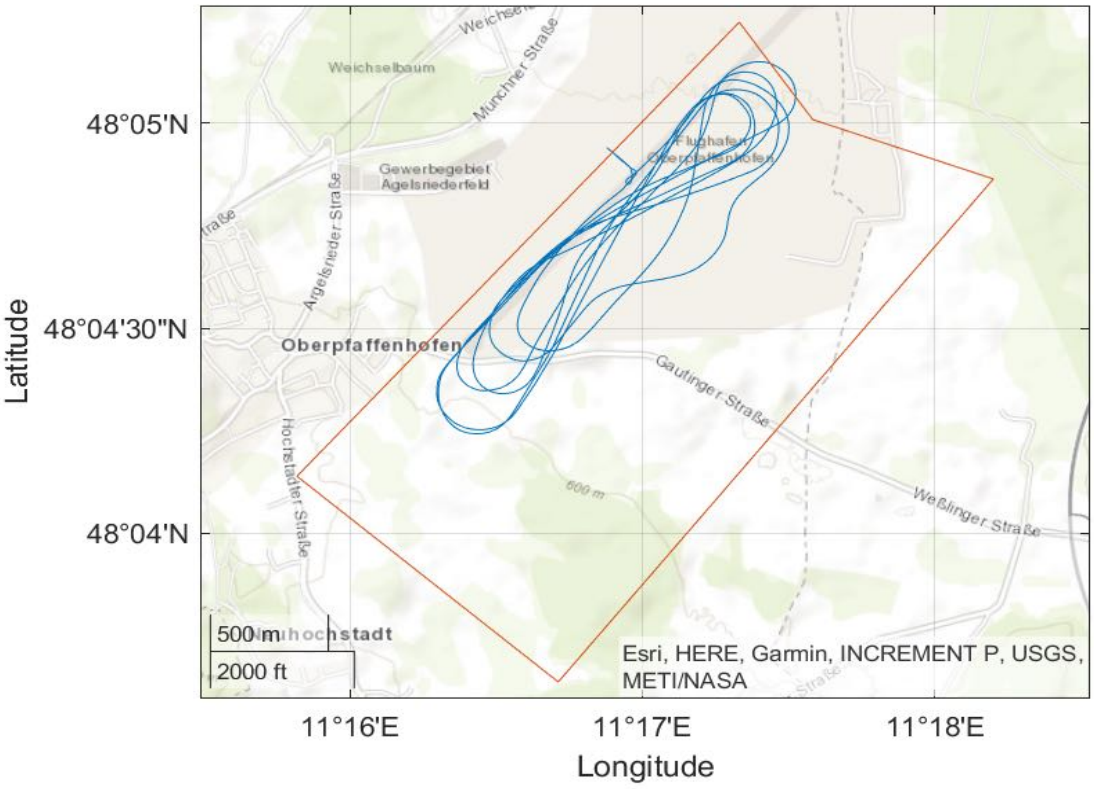


Figure 276: Flight test data

Flight Test Description for the 1st Flight Test Campaign, May 2022 In total, 8 flights were done, totalling to around 140min of flight time. The main issue was the windy conditions at the airport with big crosswind components every day. This resulted in three crashes, with one of them being especially severe. The crashes happened the day after the first flight, therefore a few days had to be spent repairing the aircraft.

After repairs it was decided to wait for a calmer weather, which did not happen during the first week. The modelled (not recorded) weather history is provided in Figure 277. In between 11-15 of May, $2-8m/s$ wind was estimated. This was also confirmed with the measured wind conditions at the two nearby weather stations (Figure 278), where $8m/s$ wind with gusts up to $16m/s$ was recorded. Note that later on it was established that the safe takeoff crosswind limit for this aircraft is $4m/s$.

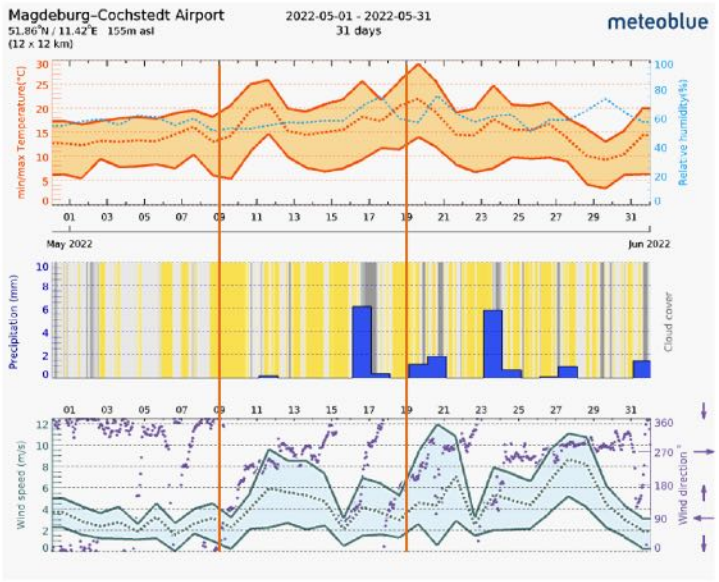


Figure 277: Modelled weather conditions during the test campaign (marked with red lines).

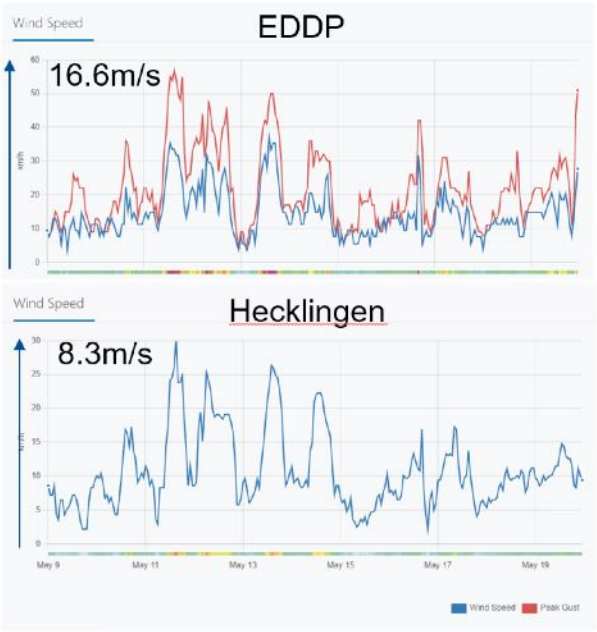


Figure 278: Measured wind conditions at during the test campaign at Hecklingen weather station (5km away from the EDBC airport) and EDDP airport (74km away from the EDBC airport).

FT10

Flight test 10 (FT10) was the first flight of the year and it was conducted in a new environment for the whole flight test crew. During the flight, the new second pilot was trained. The rest of the flight was used to check the augmented mode. At that point it was reported that the aircraft seems to slightly oscillate when the augmented mode was turned on. The flight was concluded after 13min.

Table 45: FT10 - Flight information.

Flight number:	10
Flight date:	09-May-2022 15:40:38
Take-off time:	15:45:30
Landing time:	15:58:29
Total flight time:	00:12:59
Total fuel used:	4.26 kg

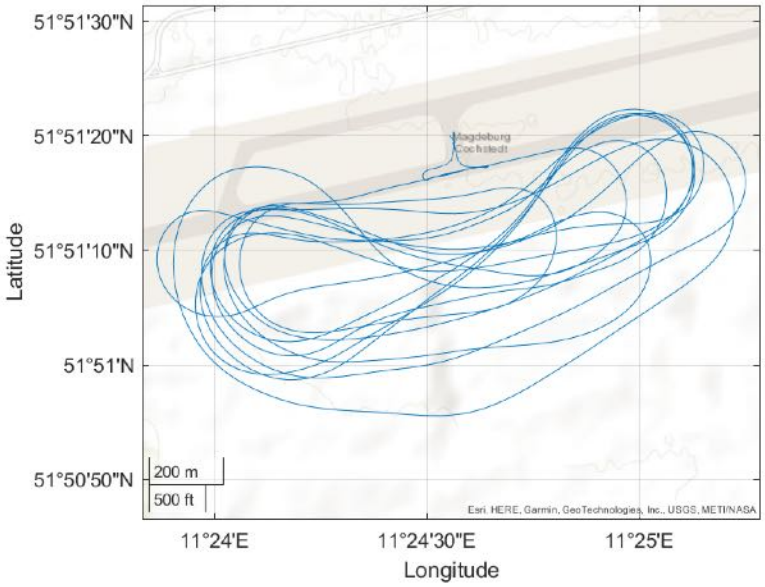


Figure 279: FT10 Trajectory plot.

Figure 280: Team photo after a successful first flight of the year.

FT11

FT11 was planned as another training flight with additional flight segments done for flexible mode identification in-flight (steady coordinated turn points). Also, for this and following flights the left wing was covered in tufts to investigate the airflow separation due to extended flaps (Figure 281). However, due to high crosswind components the aircraft veered off to the left into the wind and the wing hit a runway light while moving backwards (Figures 282 and 283). An aileron had to be changed and landing gear had to be repaired.

The repairs were done within 6 hours and another attempt for a flight was made. This time the aircraft again went to the side during the takeoff run. The damage was more serious than during the previous attempt. The fuselage was partially ripped open at the front frame, the nose boom was damaged. The parachute was ripped out (Figure 284). All the main frames were ripped out from the fuselage. Main landing gear damper was broken.

Figure 281: Left wing root covered in tufts to investigate the flow separation with extended flaps.

Figure 282: Right wing moving backwards after a ground-loop just before hitting the runway light.

Figure 283: Right wing moving backwards after a ground-loop just after hitting the runway light.

Repairing this damage was a way bigger effort, and it was not clear if it could be done in the hangar at the airfield. In the end it was decided to attempt the repair on the field. The fuselage had to be emptied from components (payload bay, cables, landing gear). The fuselage hull was then repaired

Figure 284: Tail cone flying away from the aircraft after the crash during an unsuccessful takeoff attempt.

(Figure 285) and components were reassembled. Wing alignment had to be done and centre of gravity check performed. The pitot probe had to be exchanged. All this cost another two days of the campaign, but the aircraft was ready for taxi tests within 48h of the crash.

Figure 285: Repairing the fuselage after the crash during an FT11 attempt.

After the crash further investigations were done into the takeoff performance. The flap settings were further discussed, which were increased from 23/16/5/0 for take-off to 25/16/10/5. For landing the settings were 30/16/10/5. Taking into account that full elevator down command is required during the takeoff, it was decided that maybe there is too high nose down pitching moment due to the high flap settings. Deflecting the outboard flaps down also increase the nose down pitching moment a lot due to the wing sweep. Therefore, the outboard flap setting was reduced to 25/16/10/0 for takeoff and 30/16/10/0 for landing.

After the changes another attempt was done on 14.05. But due to crosswind the aircraft went to the side again, without any damages. After this it was also decided to mount the wingtip wheels which could help with directional stability on the ground.

Finally, on 16.05 the FT11 was successfully done. Different augmented mode versions were checked in-flight. Two triangles were flown to check gather data for airspeed probe calibration. Altitude hold mode was checked, but it could not be confirmed to work at that point.

Table 46: FT11 - Flight information.

Flight number:	11
Flight date:	16-May-2022 09:40:06
Take-off time:	09:44:59
Landing time:	10:03:10
Total flight time:	00:18:11
Total fuel used:	5.75 kg

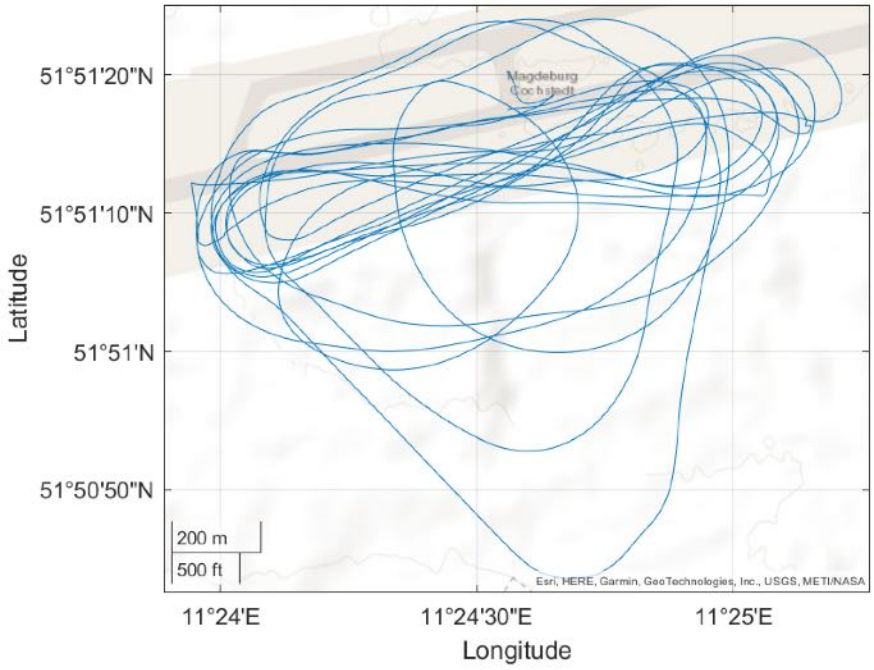


Figure 286: FT11 Trajectory plot.

FT12

On the same day as the previous flight, FT12 took place. The altitude hold mode was tested again, but due to a bug it could not be used. Therefore, only autothrottle tests were done with three different modes - robust, performance and total energy control. Two airspeed commands were issued during the autothrottle tests- 38 and 42m/s. These steps are also visible on the airspeed plot in Figure 287. At the end of the flight pilot training was done together with accelerated turn that would be used for airspeed calibration later on.

Table 47: FT12 - Flight information.

Flight number:	12
Flight date:	16-May-2022 15:42:54
Take-off time:	15:47:44
Landing time:	16:08:36
Total flight time:	00:20:52
Total fuel used:	6.34 kg

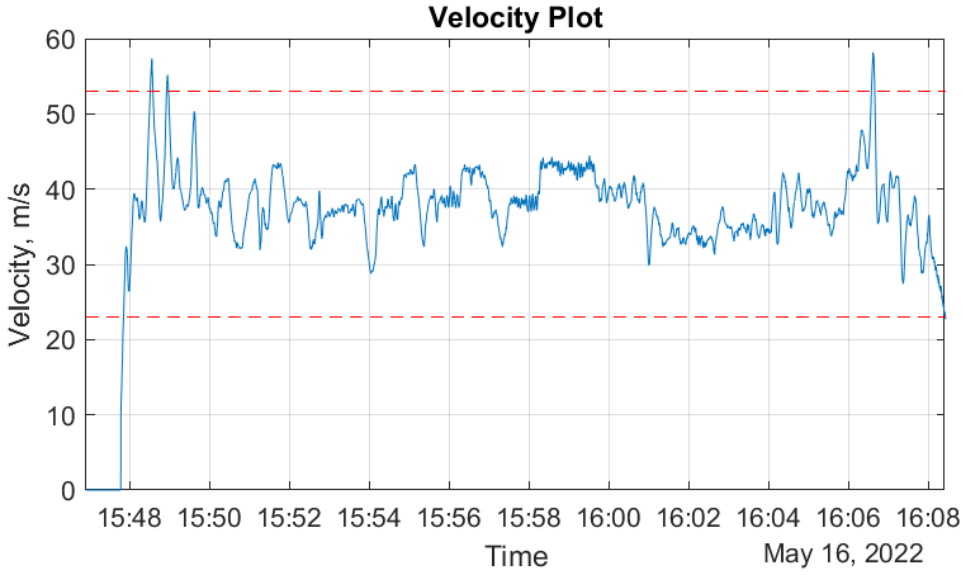


Figure 287: FT12 Airspeed plot.

FT13

Course angle hold and horse race pattern hold were tested during the FT13. Also, oil was applied on the wing root area to investigate the boundary layer in-flight. More precisely, it was attempted to locate any laminar separation that could drastically reduce lift on the wings.

Due to some problems with the autopilot software, the altitude hold mode did not work again, nor did any other autopilot module. Therefore, another triangle and an accelerated turn were flown for airspeed probe calibration investigation as a backup plan in manual mode.

Table 48: FT13 - Flight information.

Flight number:	13
Flight date:	17-May-2022 15:10:54
Take-off time:	15:15:45
Landing time:	15:36:29
Total flight time:	00:20:44
Total fuel used:	6.4 kg

FT14

One hour after landing, the FT14 was performed to again check the autopilot modes - augmented mode, altitude hold mode, autothrottle mode and course angle hold together with the horse race pattern hold.

This time, all the autopilot modes worked. Good functioning of the altitude hold mode is seen from Figure 288 (note that the wrong altitude limits are plotted). Airspeed was also tracked with reasonable accuracy using the robust autothrottle mode (Figure 289). Finally, automated horse race track pattern was flown by using all the autopilot modes for the first time. The aircraft aligned with the runway and turned around after flying straight for 400m (Figure 290). The airspeed was then decreased from 42m/s to 34m/s and the same pattern was repeated. Everything seemed to work well.

At the end of the flight there was still some fuel remaining, so manoeuvre injections for rigid body

modelling was done. Three elevator doublets, an elevator pulse and a rudder multisine input were done.

Table 49: FT14 - Flight information.

Flight number:	14
Flight date:	17-May-2022 16:52:27
Take-off time:	16:57:18
Landing time:	17:19:35
Total flight time:	00:22:16
Total fuel used:	6.17 kg

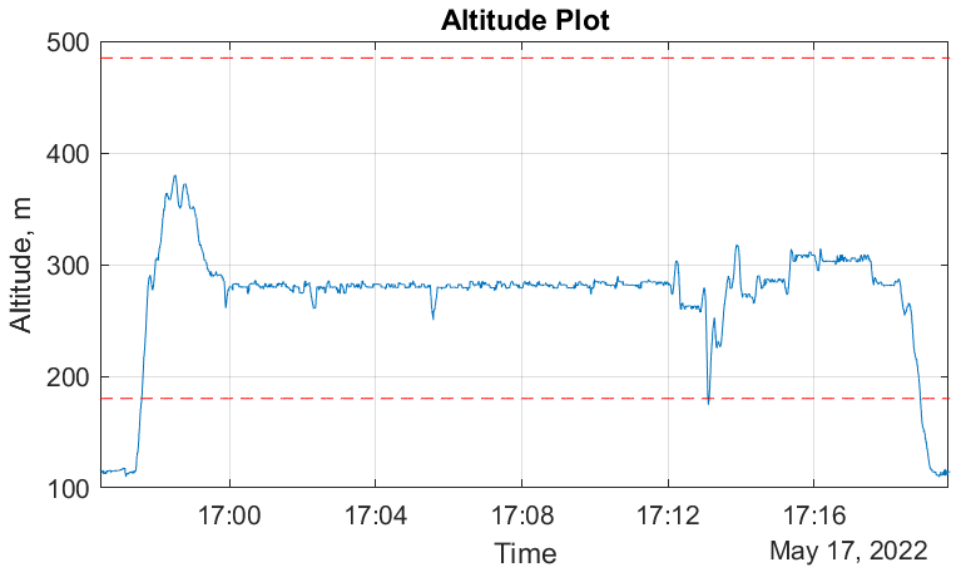


Figure 288: FT14 Altitude plot.

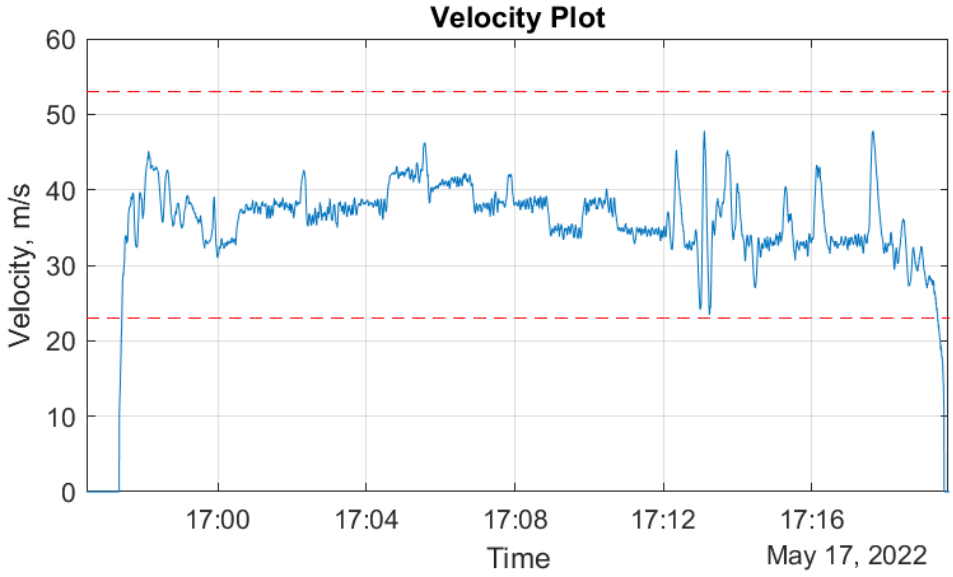


Figure 289: FT14 Airspeed plot.

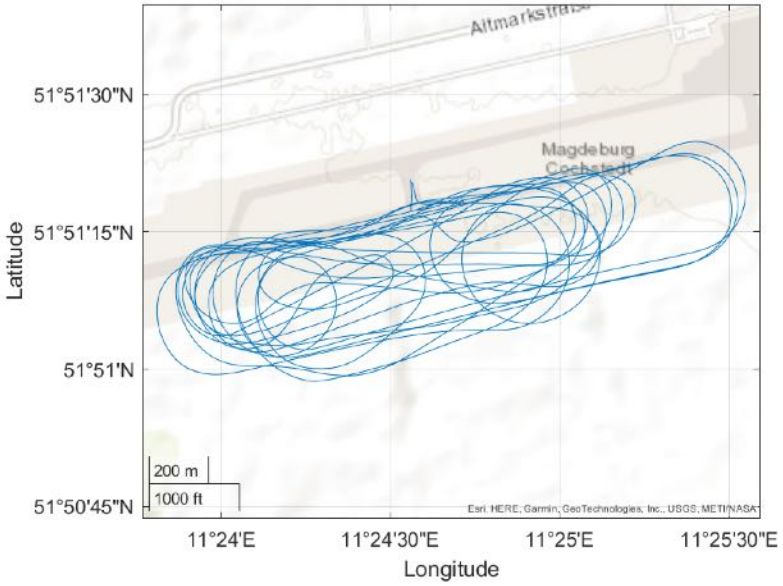


Figure 290: FT14 Trajectory plot.

FT15

Full airspeed envelope with the autothrottle mode was planned for FT15. This meant starting the autothrottle at 30m/s and then gradually increasing the airspeed up to 60m/s (the planned highest speed during flutter tests) while flying in a steady circle. However, multiple problems meant that this flight was not successful.

Firstly, the autopilot modes didn't seem to do what was commanded. While trying to solve this in-

flight, the pilot switched back to manual control and tried to increase throttle. At that time the engine was spooling up and down. The pilot tried to change the throttle setting, but the aircraft didn't react as commanded. At the same time the telemetry indicated that the aircraft is still in autopilot mode. Consequently, the pilots decided to go for landing.

During the approach the pilots realized that the aircraft was still in autopilot mode even though the command switch on the transmitter was set to manual mode. They then switched once more from manual to autopilot mode and back and aircraft control was regained.

In the end it was discovered that the problems arose due to a defect switch on the main transmitter. Furthermore, it was discovered that the flight log was not created due to changing the setting of the autopilot in-flight.

FT16

The last day of the test campaign was once again windy, so the flights had to wait for the evening. Same goals were assigned for FT16 as for FT15. This time the autopilot modes worked well and the full airspeed envelope of the aircraft was checked. It was also noticed that the airspeed was kept lower than the commanded value. The turns were done automatically up to 50m/s. After that they had to be done manually due to the saturation of the autothrottle command. The complete airspeed graph is plotted in Figure 291.

Finally, for the last part of the flight, two segments were flown with the so called "drag flap" state. This configuration was purposefully designed to increase induced drag. It was hoped that this way it could be established if the drag measurement algorithm can detect an increase of induced drag in-flight.

Table 50: FT16 - Flight information.

Flight number:	16
Flight date:	19-May-2022 16:09:56
Take-off time:	16:14:46
Landing time:	16:36:31
Total flight time:	00:21:45
Total fuel used:	6.54 kg

A description and analysis of the flexible mode identification using data from Flight Test 16 is added in Section 2.3.3.

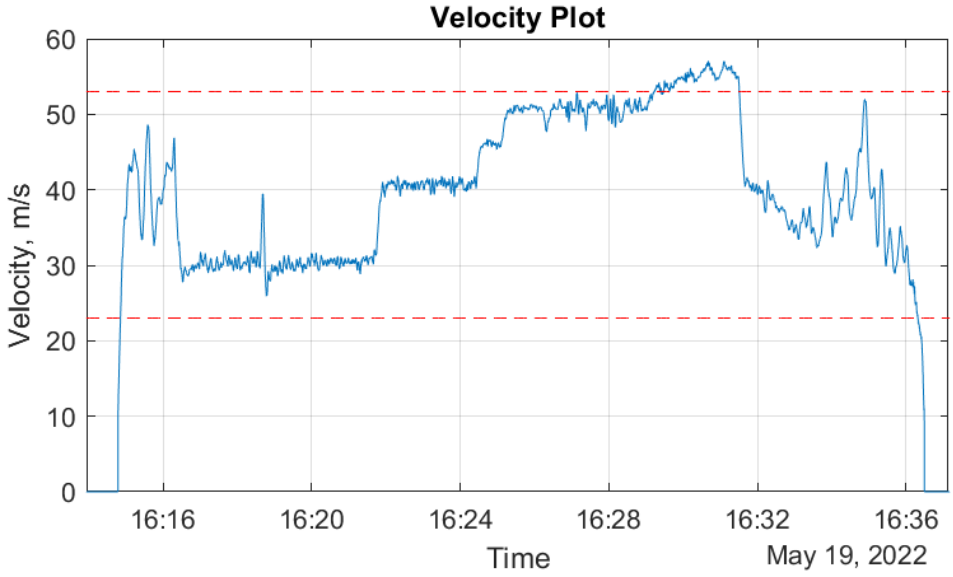


Figure 291: FT16 Airspeed plot.

FT17

Within less than an hour, the second flight of the day, FT17, was done. The flight was targeted at rigid body mode identification manoeuvres - elevator multisine inputs, elevator pulses, aileron multisine inputs and rudder doublets. In total, 14 manoeuvres were done. During these automated inputs it was noticed that the aircraft is highly unstable in roll, and any gust makes it turn to the side. This was the reason why during all of these manoeuvres the aircraft didn't fly completely straight.

During the flight, the second pilot mentioned that the wind is rising. After a second warning it was decided to go for landing, as the wind speed was steadily increasing. This decision was made too late and the aircraft had to be landed in stormy conditions with wind speeds rising from 1 to 7.1m/s and gusts from 1.5 to 15.4m/s. Luckily, the landing went well and the first flight test campaign of the year could be concluded.

Table 51: FT17 - Flight information.

Flight number:	17
Flight date:	19-May-2022 17:15:13
Take-off time:	17:20:05
Landing time:	17:41:42
Total flight time:	00:21:37
Total fuel used:	5.5 kg

Flight Test Description for the 2nd Flight Test Campaign, August 2022 The flight test crew arrived back at EDBC on 19th of August for the 2nd Flight Test Campaign. But due to the upgrades on the aircraft, the first three days were spent on debugging various issues. It was noticed that the tailwheel steering is coupled to a wrong channel (tail flap deflection instead of a rudder command). Also, the air data system didn't work properly. Only after careful inspection it was realised that the tubes inside the pitot boom are tangled, and the whole system was retubed. Then, during a taxi test the aircraft touched a runway light, which meant that a small repair had to be undertaken. Autopilot modules also ahd to be

adjusted based on the ground tests and some further tweaking of the tailwheel steering had to be done after the taxi tests on 22nd of August. Finally, two wing servos had to be replaced due to them burning down during ground tests before continuing with the flight tests the next day.

FT18

Flight test 18 was the first flight test of the second flight test campaign at EDBC. The first test was all focused on pilot training. Three landing imitations were done for the new main pilot to get familiar with the aircraft before landing was done. Only correct functioning of the augmented autopilot mode was checked in-flight. Also the fuel flow meter sensor did not work with the new hardware, so the flight times had to be limited for safe operations.

Note that the take-off and landing flap settings were again changed for this flight based on the takeoff performance investigation made after the 1st flight test campaign. The outboard flaps were deflected upwards to provide more pitch-up moment and help to push the tail down during the ground run, as further described in Section 2.3.3.

Table 52: FT18 - Flight information.

Flight number:	18
Flight date:	23-Aug-2022 09:15:09
Take-off time:	09:20:01
Landing time:	09:37:38
Total flight time:	00:17:36
Total fuel used:	(fuel sensor malfunction)

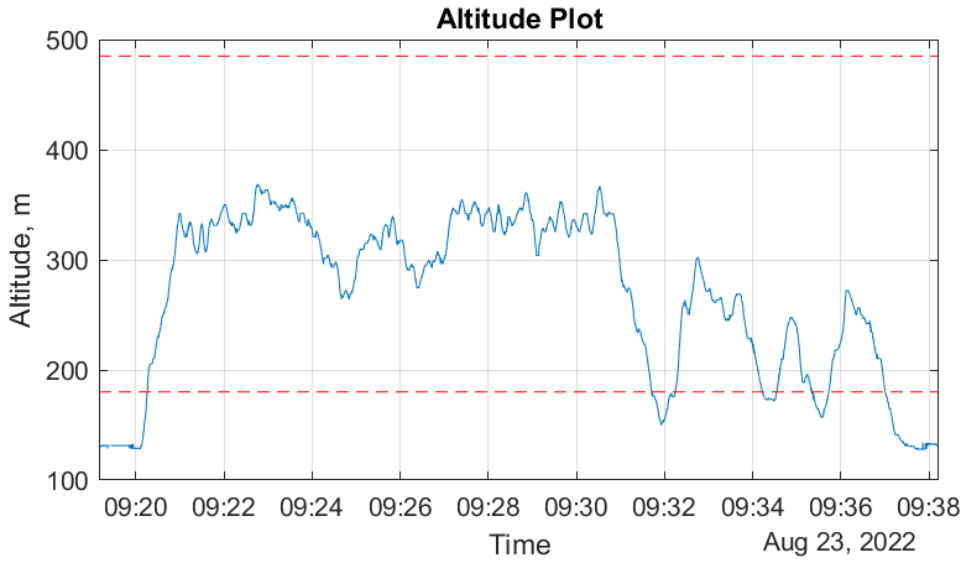


Figure 292: FT18 Altitude plot.

FT19

Flight test 19 was carried out on the same day as flight test 18. Further tests of the autopilot functionalities were done. This time altitude hold mode was checked. The first try didn't seem to work. The aircraft pitched up a lot when the altitude hold mode was triggered. The next try seemed to work better, but not as good as during the last campaign and it was hard to say what will the aircraft do when the altitude hold will be switched on. Autothrottle mode seemed to work well, smoother than previously.

Two steps in commanded airspeed were done for two autothrottle models (robust and performance) and both seemed to work reasonably well.

After the autopilot tests, three pushover-pull-up manoeuvres were done.

Table 53: FT19 - Flight information.

Flight number:	19
Flight date:	23-Aug-2022 15:36:52
Take-off time:	15:41:30
Landing time:	15:58:13
Total flight time:	00:16:43
Total fuel used:	(fuel sensor malfunction)

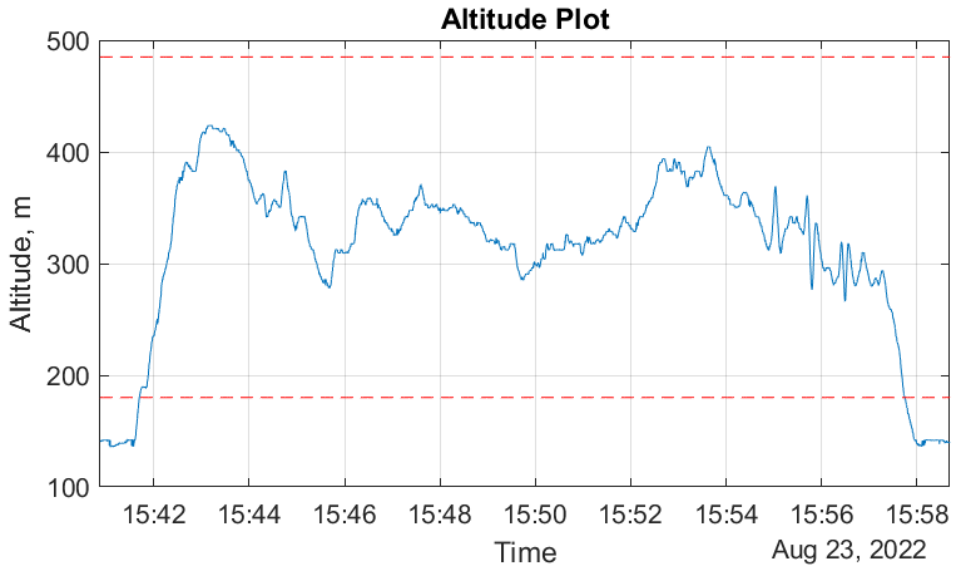


Figure 293: FT19 Altitude plot.

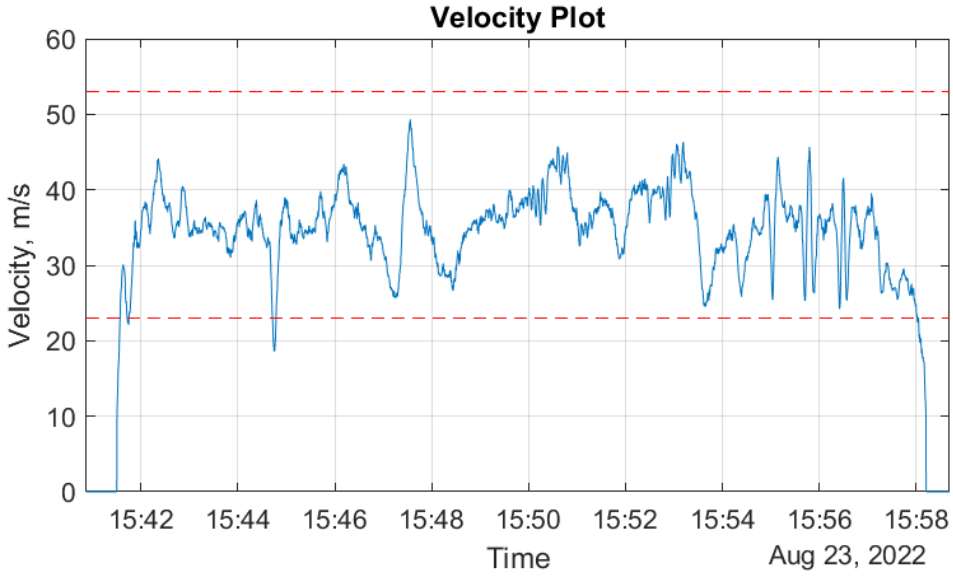


Figure 294: FT19 Airspeed plot.

FT20

Reception problems influenced the flight test 20 a lot. The main telemetry link was barely usable in-flight, so the secondary link had to be used to report airspeeds to the pilots. Five pushover-pull-up manoeuvres were done at load factors from 1.9G to 3.1G. Tailwheel steering servo gear was broken during landing and had to be replaced.

During this flight the fuel flow meter telemetry started working again. Therefore, the consumed fuel could be tracked in-flight and the flight times could be increased again, if not for the incident.

Table 54: FT20 - Flight information.

Flight number:	20
Flight date:	24-Aug-2022 10:10:39
Take-off time:	10:15:30
Landing time:	10:31:30
Total flight time:	00:15:59
Total fuel used:	5.1 kg

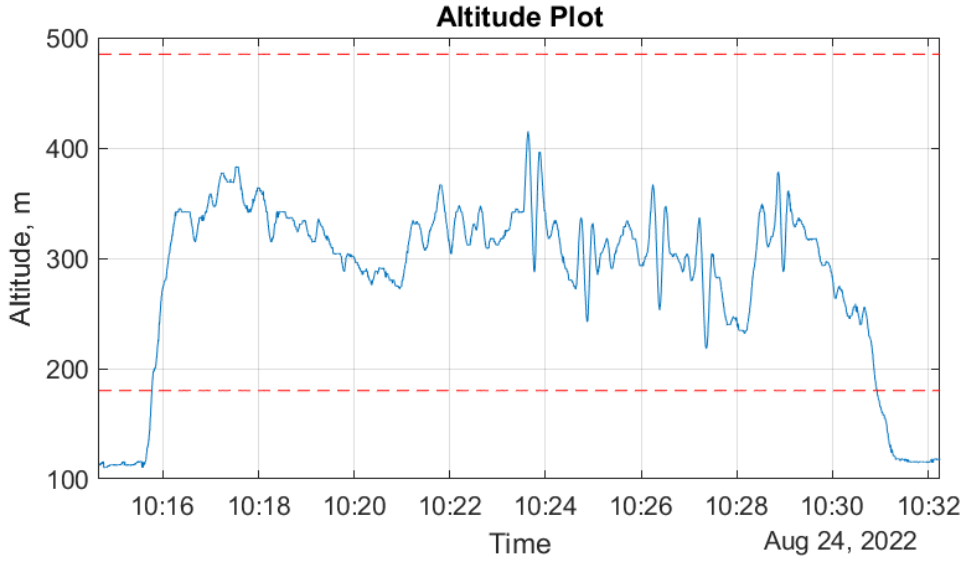


Figure 295: FT20 Altitude plot.

FT21

Flight test 21 focused on rigid body mode identification manoeuvres. Six manoeuvres were done. All of them seemed to be unstable and the aircraft rolled sideways during the manoeuvre injection. It was also noticed that the augmented mode commands very big rudder deflections to keep the aircraft at a zero sideslip angle. Later on while reviewing the on-board videos it was recognised that these rudder deflections would result in complete tail twisting. The problem was sent for further investigation.

As the pilot was preparing for another manoeuvre injection the engine flamed out (Figure 297). Luckily this happened at a safe altitude and in a convenient position. Expedited landing checklist was initiated and the secondary pilot helped the main pilot with safely gliding the aircraft down. It was later on discovered that a faulty tube connector allowed air to enter the fuel system and the engine was not getting enough fuel anymore. The problem was realised while investigating the 360 degree camera videos where the air bubbles are visible in the fuel line running to the engine. The faulty connector was replaced for the next flight.

After the event, an engine flame-out checklist was prepared.

At this point it was also noticed that pushing the tail down immediately after landing helps with ground controllability.

Table 55: FT21 - Flight information.

Flight number:	21
Flight date:	24-Aug-2022 12:55:38
Take-off time:	13:00:30
Landing time:	13:15:56
Total flight time:	00:15:26
Total fuel used:	4.95 kg

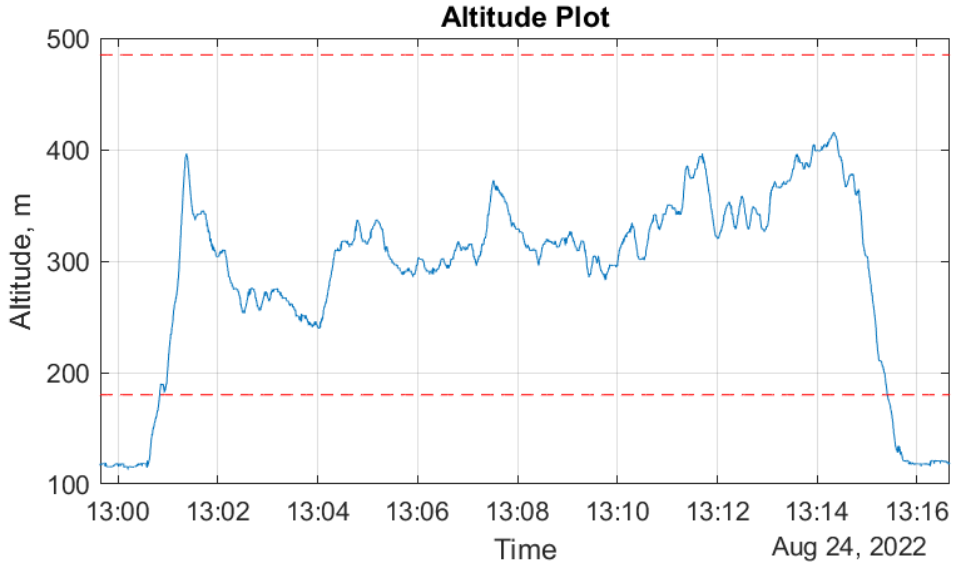


Figure 296: FT21 Altitude plot.

Figure 297: Engine flameout during the FT21.

FT22

Flight test 22 was supposed to further focus on rigid body mode identification manoeuvres. However, 4 minutes after take-off the engine has shut down once again. Luckily, the aircraft had enough altitude and speed for the pilot to carry out a landing. Engine flameout checklist was followed and the aircraft was once more brought down safely.

The problem was later on attributed to a bug in the flight control computer which, at a rare instance, could issue a split-second command for the engine to turn off.

Table 56: FT22 - Flight information.

Flight number:	22
Flight date:	29-Aug-2022 16:18:03
Take-off time:	16:22:55
Landing time:	16:26:15
Total flight time:	00:03:20
Total fuel used:	1.21 kg

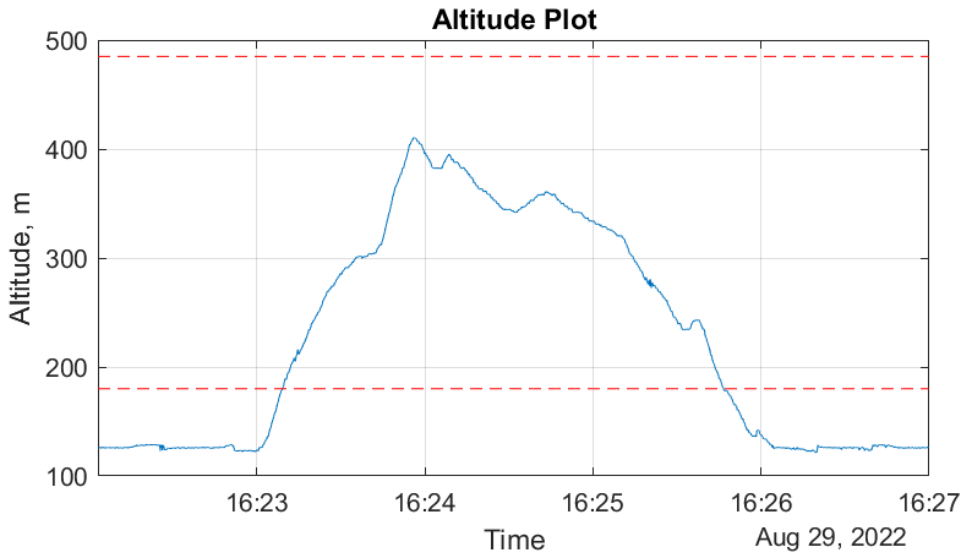


Figure 298: FT22 Altitude plot.

FT23

Flight test 23 was once more planned for rigid body mode identification. The flight was going well, with the telemetry looking stable and the weather being comfortable for flying as well. Five manoeuvres were done - three elevator multisines with an increased amplitude and two aileron multisine inputs.

During the flight there were no red flags regarding a bad reception. The distances in between the pilot and the aircraft looked good from both optical and telemetry quality perspectives.

Around nine minutes in-flight, the pilot was performing a right turn at the eastern part of the flight box (Figure 299). He saw some white trace following the aircraft and thought that the engine has flamed out once again. Thinking he is still in control, he tried to turn the aircraft back. At that moment the pilot realised that it was the the parachute that was coming out from the aircraft. He announced that via the intercom and the crew followed the ”Terminate” checklist which was prepared in such case. Both pilots were commanded to initiate the TERMINATE command, which then also shuts the engine down.

As the pilots were returning to the ground control station, the Flight Test Manager has informed the tower about the aircraft coming down with a parachute. It was clearly visible to the tower. Within a minute the aircraft hit the ground, outside of the airport zone. After a while the tower has confirmed that a fire has broken out at the impact location. At that point two cars, including the ground control station were driving to the crash site.

Eight minutes after the impact both cars reached the crash site. At that point, only a small fire remained at the front part of the fuselage. The fuselage section from the wing root to the tail was completely burnt

down. The wing roots were also burnt. The parachute itself was found laying further back of the aircraft. The remains of the aircraft were extinguished. Extra effort was made to recover as much flight data as possible, including the cameras, to investigate the accident. Luckily, no one got hurt during the crash and the field was mowed down, therefore the fire was not being fueled.

The crash investigation report can be found in Section 2.3.3.

Table 57: FT23 - Flight information.

Flight number:	23
Flight date:	30-Aug-2022 16:24:55
Take-off time:	16:29:47
Landing time:	16:39:19
Total flight time:	00:09:32
Total fuel used:	2.96 kg

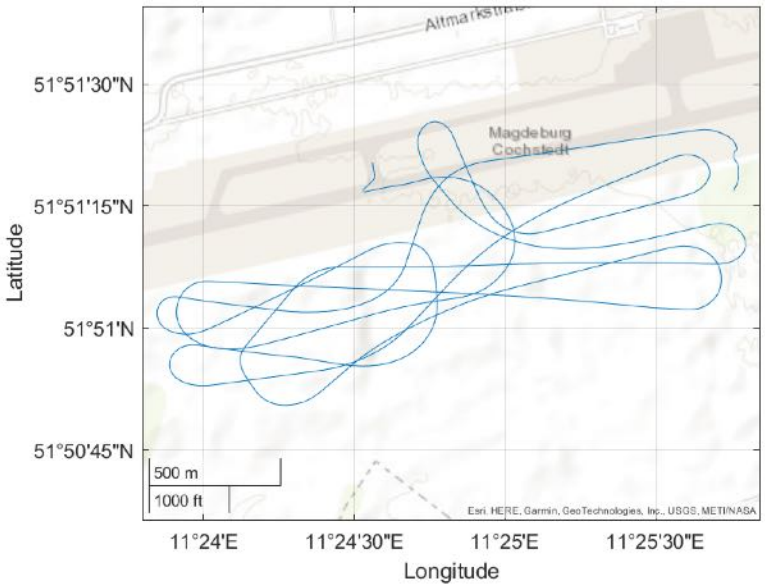


Figure 299: FT23 Trajectory plot. The end of the trajectory in the eastern part of the map is where the aircraft landed with the parachute.

Conclusion of 1st and 2nd flight test campaign During the two flight test campaigns in 2022, 14 flights were made with the baseline demonstrator. Most of the flights averaged around 16 minutes and more than 220 minutes of flight time was collected. The report can be concluded with the following points:

- The aircraft's ground controllability (and, therefore, reliability) was significantly improved. A severe crash was experienced during the take-off in FT11, but upgrading the landing gear once more (with the wingtip wheels) and changing the flap configuration solved the issues that were seen as the biggest challenges for flight testing of this demonstrator.
- New pilot was trained, which increased the flexibility of the flight test team. Furthermore, the flight

test team had further optimized flight testing operations, resulting in better preparation for the second test campaign.

- Data was gathered for baseline aerodynamic evaluation of lift and drag, summarised in a publication. Additionally, data was gathered for identifying the flexible body modes in flight.
- All the autopilot modes were tested and proven to work in-flight with only minor updates required before the flutter flights can commence. The autothrottle mode, which appeared to be the most challenging mode to adjust, was tested up to $56 m/s$.
- Airspeed calibration was confirmed to be within the required limits using i-flight data.
- The crash during FT23 was investigated in detail, which touched the points that were not investigated before.

Only some of the planned goals were achieved during the campaigns. For example, the flexible -1 wing still needed to be flown. This was planned for the third campaign of the year, but could not be undertaken due to the crash. Additionally, the manoeuvres injected for rigid body mode identification were insufficient to perform proper data analysis.

Flight Test Description for the 3rd Flight Test Campaign The first week of the 3rd Flight Test Campaign did not result in any flight tests. On March 27th and 28th the wind, temperature and precipitation did not permit to go outside with the aircraft (Figure 300). Engine tests had to be done outside on March 29th and further debugging done. Wind increased again on March 30th and rain did not allow for any flights on March 31st.

Only slightly better weather conditions were experienced during the second week. The wind was again above the limit on 3rd of April (Figure 301) and only April 4th and 5th were actually flyable days. During those days, two flights were completed (Figure 302).

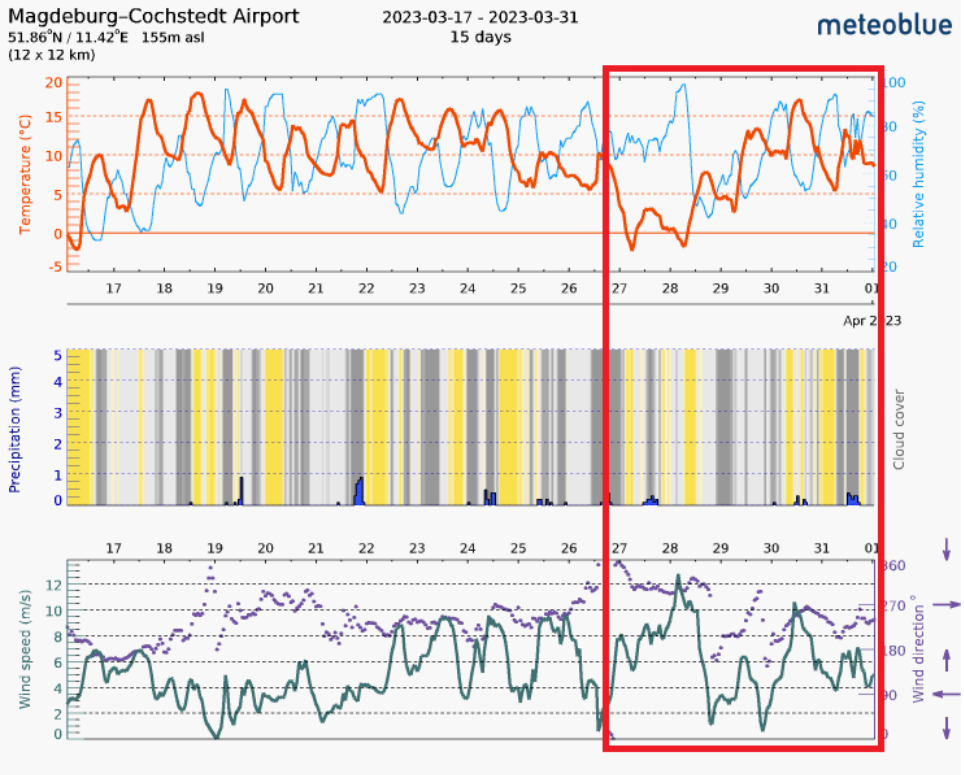


Figure 300: Weather conditions at the EDBC airport for the 1st week of the 3rd Flight test Campaign. The days when the campaign took place are marked in red.

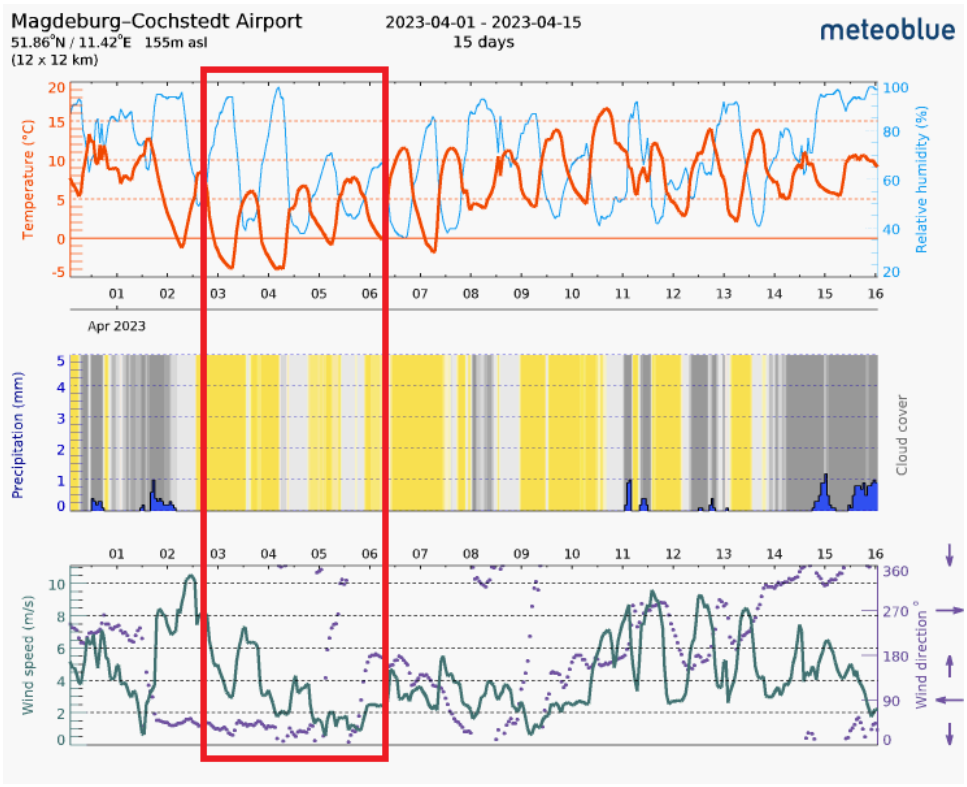


Figure 301: Weather conditions at the EDBC airport for the 2nd week of the 3rd Flight test Campaign. The days when the campaign took place are marked in red.

Figure 302: Preparing the aircraft before the flight test 24.

FT24 - Maiden flight of the new demonstrator

The maiden flight of the new demonstrator took place on 04.04. The goal of the flight was to confirm that all the systems work in-flight, confirm that the controllability of the flexible wing is the same as for the old demonstrator, perform landing imitations as part of the pilot training and do the required autopilot mode checks.

Table 58: FT24 - Flight information.

Flight number:	24
Flight configuration:	Flutter-safe
Flight date:	04-Apr-2023 12:12:54
Take-off time:	12:17:51
Landing time:	12:36:56
Total flight time:	00:19:04
Total fuel used:	6.41 kg

After the take-off, trim settings were confirmed. Secondary air-data probe readings were also compared to the readings of the main air data system. The secondary probes showed $3 - 4m/s$ higher than the main probe. A calibration curve for the secondary probes was created later on.

A few minutes in-flight a high voltage drop in one of the batteries was noticed. It was decided that the voltage is still within the pre-set limits. During the landing imitation training it became clear why did the voltage drop occur. One of the inner flaps was not deployed in landing configuration, meaning that its servo motor has probably burnt down (Figure 304). It was decided to continue in cruise configuration for the rest of the flight.

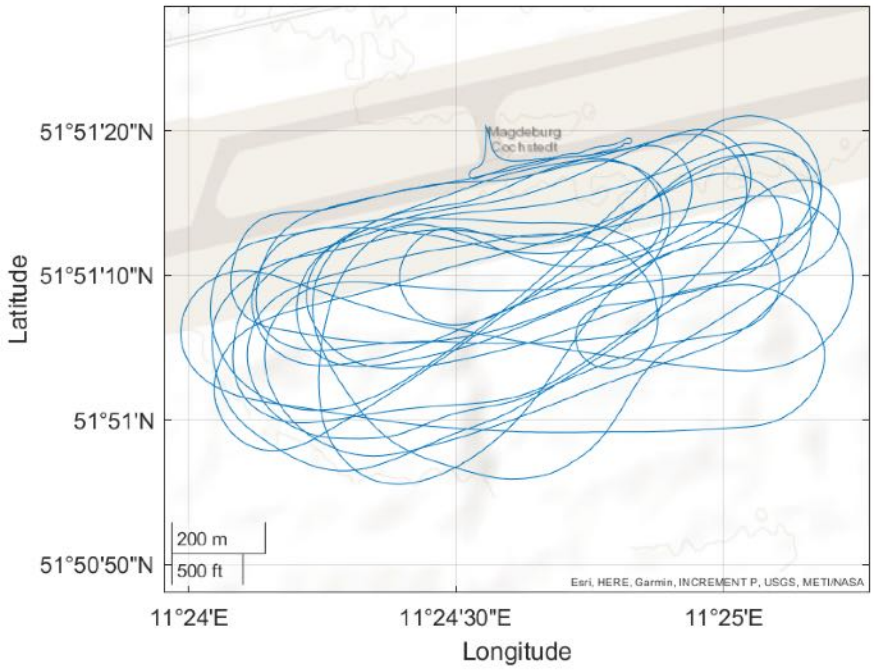


Figure 303: FT24 Trajectory plot.

Next, autopilot modes were checked. Augmented and altitude hold modes were checked in-flight. At

Figure 304: Left inboard flap is seen inactive during flight with landing configuration.

that moment a low voltage level warning for the battery, which experienced the drop before, appeared. It was decided to end the flight there and the pilots have guided the aircraft for landing. Landing, even in cruise configuration, was smooth.

During the flight telemetry quality was followed. The Mission Planner link had a reasonably good data rate, but the Engineering Data Link had a data rate of around 2 packages per second (both links had reception of more than 10Hz in the laboratory conditions). It was a point that still had to be investigated.

After the flight, a signature ceremony took place where everyone attending the test campaign had to sign the fuselage of the new demonstrator.

Later on during the day it was confirmed that the voltage drop appeared because of the burnt servo motor. In that case, a high current draw is experienced until the point where the servo motor is completely destroyed. That is where the voltage drop in the first part of the flight came from. When using the flight test data from this flight it has to be noted that roughly 80 percent of the flight was done with an unusable and uncontrolled inboard flap on the left.

During this flight it was also confirmed that the auxiliary fuel system worked, pumping the fuel from the auxiliary to the main tank in-flight.

FT25 auto-throttle, course angle hold and manoeuvre injection tests.

After exchanging the burnt flap servo and confirming the rest of the systems functional, a second flight was prepared for the next day. The goal of the flight was to finish the autothrottle mode tests and check the course angle hold mode. After that - do manoeuvre injections.

In-flight, all three autothrottle modes were checked. Due to a human-error, the first autothrottle mode was tested without altitude hold. It worked for airspeed commands 38 and 42 (within $2m/s$ accuracy). Following tests were all done with the altitude hold mode turned on. It was noticed, however, that altitude tracking was nto as good as previous year.

Then the tests moved on to course angle hold mode. After turning it on the aircraft was supposed to follow the heading in parallel to the runway, but this did not work as expected.

Autopilot's ability to follow a coordinated turn with a specified angle was then tested. During the test it

Table 59: FT25 - Flight information.

Flight number:	25
Flight configuration:	Flutter-safe
Flight date:	05-Apr-2023 12:41:48
Take-off time:	12:46:47
Landing time:	13:25:14
Total flight time:	00:38:27
Total fuel used:	9.86 kg

seemed like the actual turn radius was bigger than commanded radius of $400m$.

At some point during the autopilot tests the airspeed started to oscillate from 30 to 38 to $42m/s$. It also seemed like the altitude was increasing. At the point where the oscillations crossed the expected limits, the pilots switched back to manual mode. After this incident the autothrottle and altitude hold modes were disbled for the remainder of the flight.

Next, ten flexible mode manoeuvres were done - symmetric and asymmetric chirp signals on the outboard flaps, two types of doublets on the second, third and fourth flap.

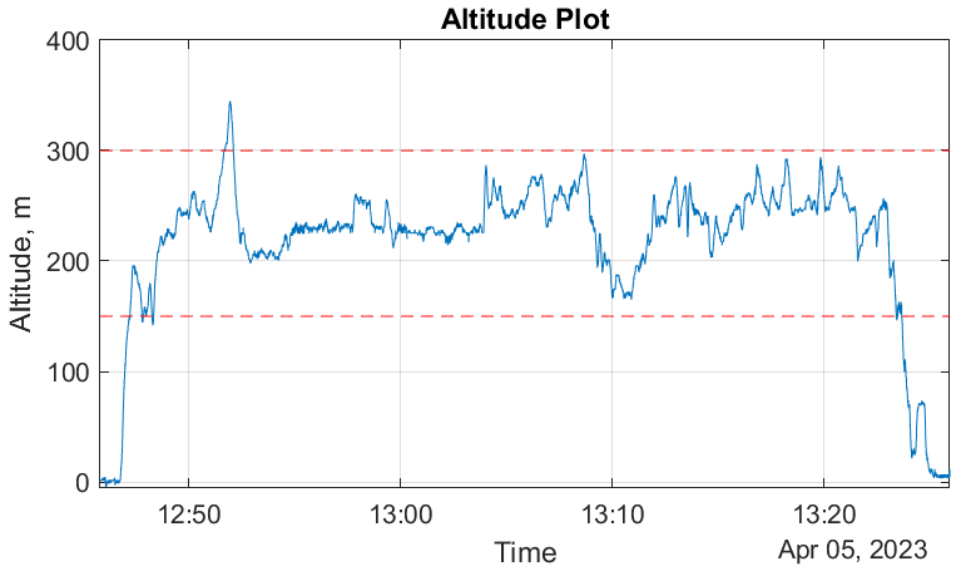


Figure 305: FT25 Altitude plot.

At the point where $8.7kg$ of fuel was used, the pilots went for landing (it was planned to go for landing with not more than $8.8kg$ of fuel used). The first approach was too fast and a go-around was issued. During the touchdown of the second flight the aircraft went slightly sideways and did a ground loop. Sadly, this resulted in breaking the tailwheel assembly and the wingtip wheels. There was no way to repair it in time for the second flight that day. Therefore, the test campaign was concluded with the flight test 25.

The implementation of the auxiliary fuel system proved its worth during the flight test 25, which was by far the longest flight test ever done in projects FLEXOP or FLIPASED - 38 minutes and 27 seconds. The previous longest flight of the old demonstrator was 22min and 16 seconds.

Flight Test Description for the 4th Flight Test Campaign 12 Flight tests were done during the 4th and final flight test campaign of the project. Starting with updating the autothrottle controller the tests became more and more complex. In the end they culminated in passing the flutter boundary in a closed loop as well as witnessing it in an open loop. More than 50 manoeuvers were done for identification and other test points that still needed to be analyzed.

Even though the weather conditions for the campaign were better than in April, they were still not ideal. High crosswinds dominated the first part of the first week, and rain took over for the second half of the week (Figure 306). Nevertheless, five flights were made that week.

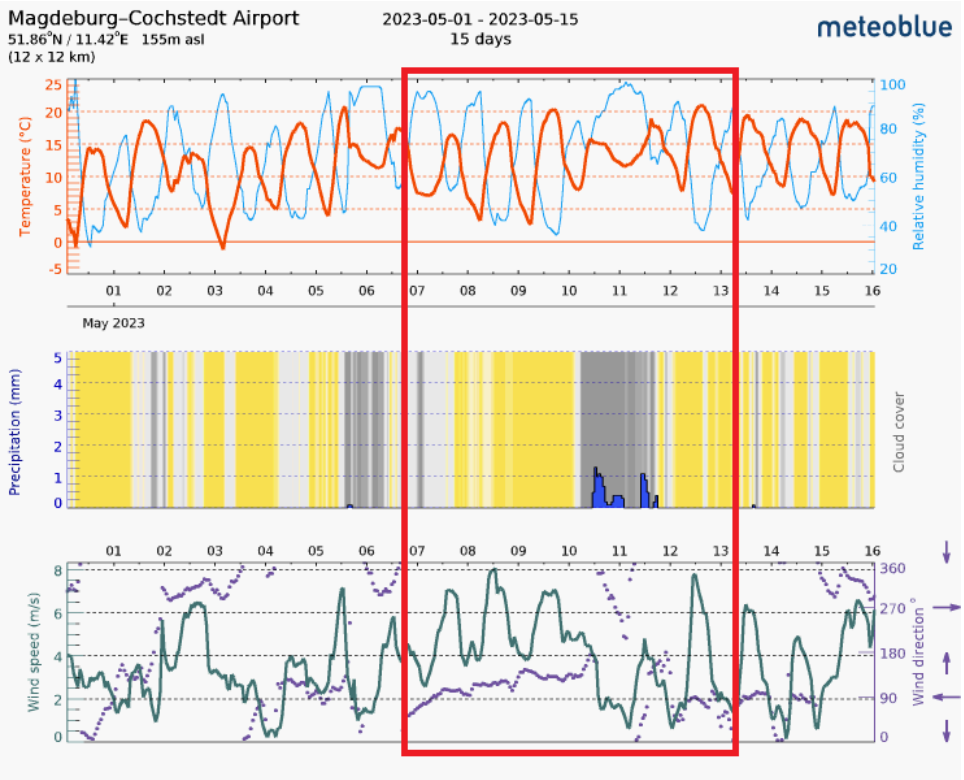


Figure 306: Weather conditions at the EDBC airport for the 1st week of the 4th Flight test Campaign. The days when the campaign took place are marked in red.

The third week was better (no flights were required the second week), with only 23rd of May being completely unflyable with more than $8m/s$ cross-wind (Figure 307). Note that the previous cross-wind limit was set to $4m/s$.

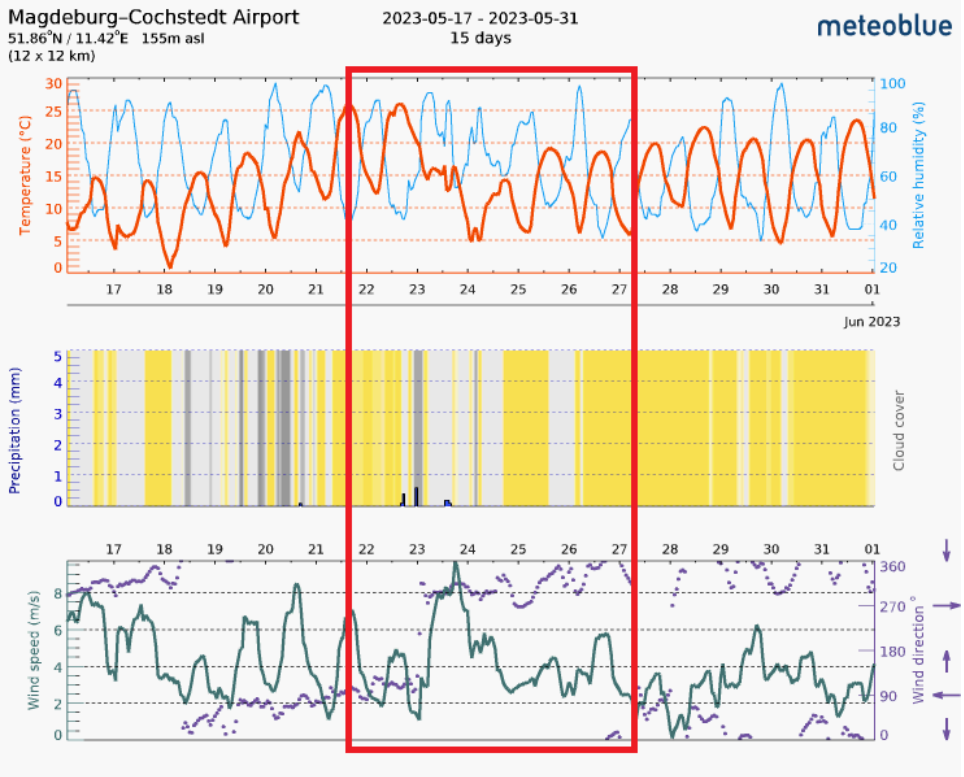


Figure 307: Weather conditions at the EDBC airport for the 2nd week of the 4th Flight test Campaign. The days when the campaign took place are marked in red.

FT26 Autothrottle, coordinated turn and manoeuvre injection tests

The plan for the flight test 26 was to do further checks on the autothrottle controller and also investigate how does it perform when used together with the coordinated turn module. Furthermore, manoeuvres from block 1 were planned.

The altitude hold mode worked well (see Figure 308). Three different autothrottle controllers were checked. Airspeed commands were varied in following steps: 38, 42, 34, 50, $38 m/s$ (clearly visible in Figure 309). It was concluded that two of the three controllers (Robust 1 and Performance) tracked the airspeed accurately within $1 - 2 m/s$ error of the commanded value. Coordinated turn mode seemed to work, but the autothrottle didn't seem to function together with that mode.

Table 60: FT26 - Flight information.

Flight number:	26
Flight configuration:	Flutter-safe
Flight date:	09-May-2023 07:02:25
Take-off time:	07:07:18
Landing time:	07:38:03
Total flight time:	00:30:45
Total fuel used:	9.81 kg

The Mission Planner telemetry link antennas were changed on the aircraft side for this flight. However, there were many issues with the telemetry during the flight. Mission Planner was not as good as with

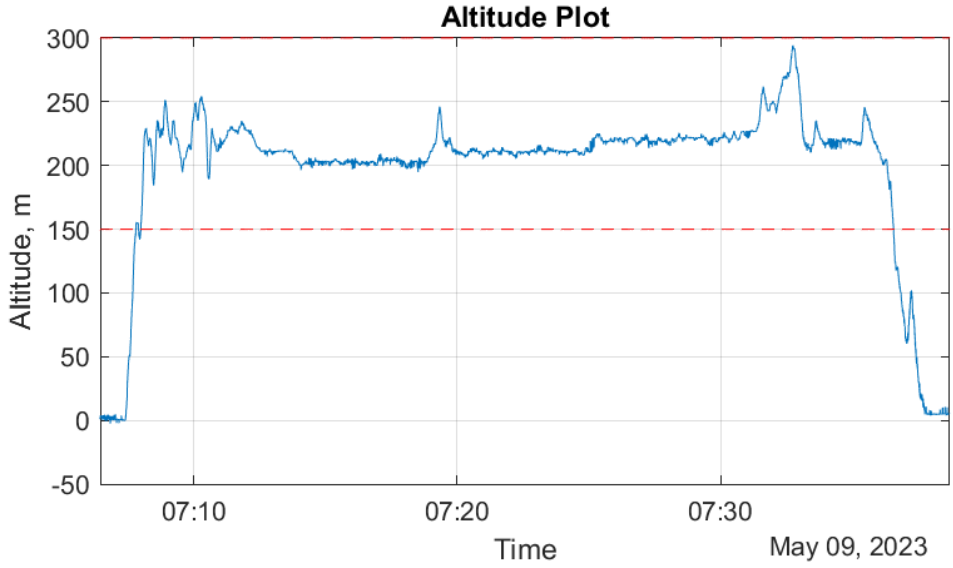


Figure 308: FT26 Altitude plot.

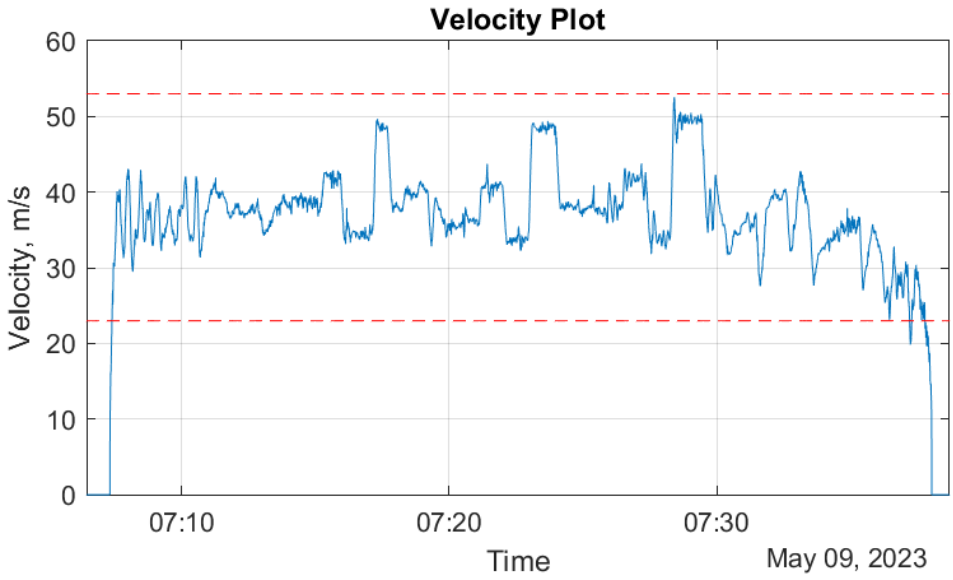


Figure 309: FT26 Airspeed plot.

the previous demonstrator, and the reception quality of the Engineering Data Link was noted as ”very bad”. Further improvements were required before the flutter tests can take place, because critical data has to be evaluated on the ground during flight.

These issues were investigated the next day and multiple solutions were proposed. However, the biggest recognition was the interaction in between the telemetry links and the long range intercom radio. After this was realised, different channels of the intercom radio were tested and it appeared that everything, but the default channel, resulted in better reception in both the intercom and the telemetry. As the itnercom radios were never used in lab, it was clear why this interference was not noticed before.

FT27 Course angle hold, autothrottle and manoeuvre injection tests

Flight test 27 was carried out two days later (the day in between had unflyable weather). The autothrottle was supposed to be tested throughout the complete airspeed envelope (up to $52 m/s$) and the manoeuvres from block 1 were prepared.

Table 61: FT27 - Flight information.

Flight number:	27
Flight configuration:	Flutter-safe
Flight date:	11-May-2023 09:21:36
Take-off time:	09:26:28
Landing time:	10:00:23
Total flight time:	00:33:54
Total fuel used:	10.7 kg

This time the autopilot modes worked well together. Autothrottle controller (from now on only Performance controller will be used) required some time to adjust to the initial command (see beginning of the flight, Figure 311), but after that it functioned well. So did altitude hold mode (see first part of the flight, Figure 310). Finally, the two modes were combined with the coordinated turn mode and turns were flown by autopilot while controlling the altitude and the airspeed (see the circles in the map, Figure 312). This was the first time that all the autopilot modes, which were required for flutter tests, functioned together.

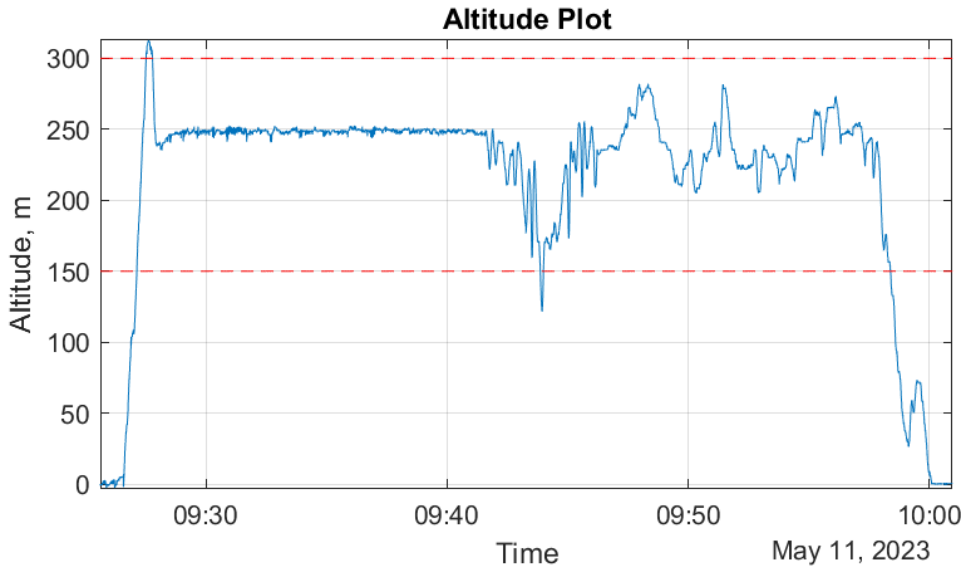


Figure 310: FT27 Altitude plot.

During the second part of the flight, three pushover-pull-up manoeuvres were done, followed by multisine injections on the elevator, rudder, flaps 2-3-4 and ailerons, as well as symmetric and asymmetric chirps. In total, 14 manoeuvres were done.

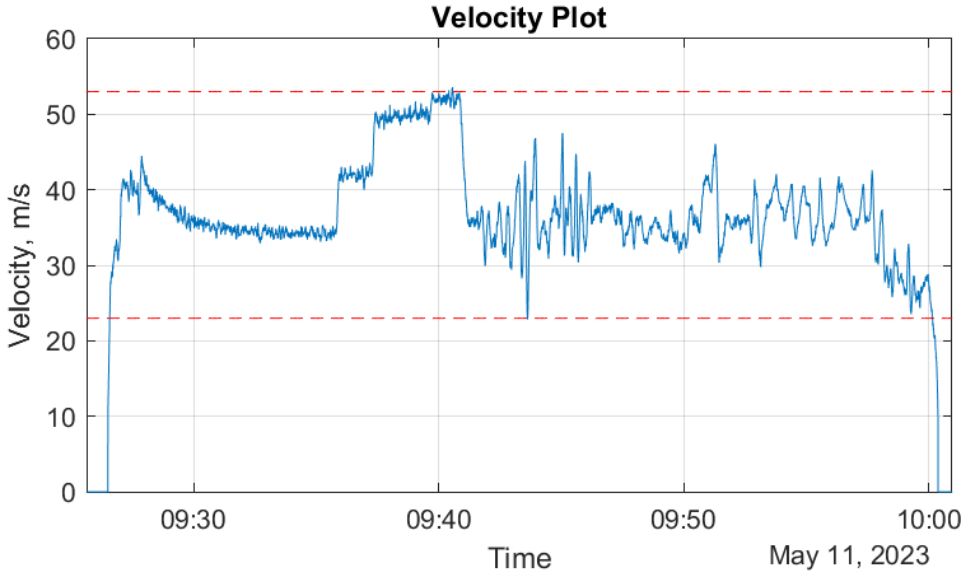


Figure 311: FT27 Airspeed plot.

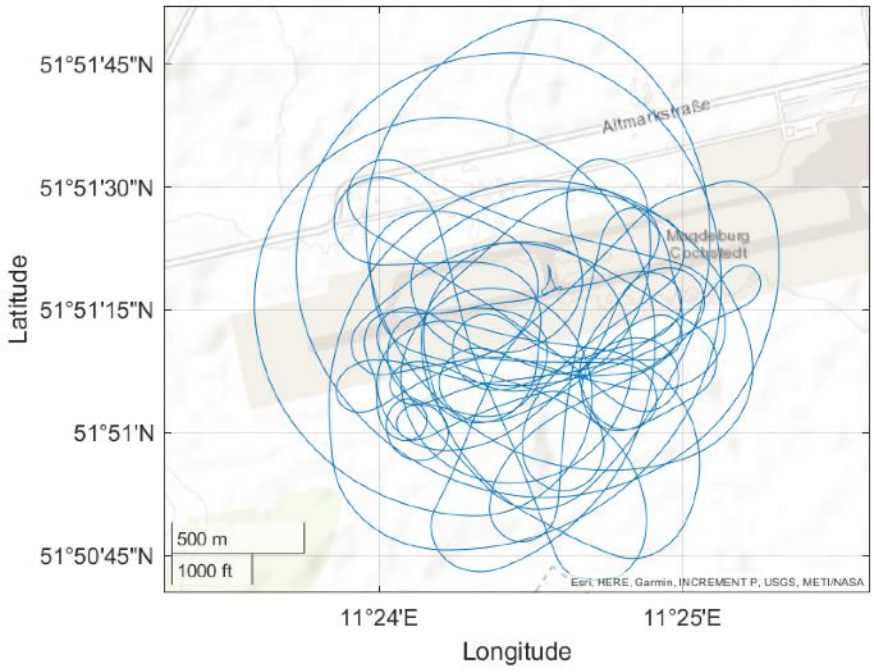


Figure 312: FT27 Trajectory plot.

FT28 Maneuver injection tests

Just one and a half hour after the FT27, the next flight was done. This time the complete flight was planned for manoeuvre injections, blocks 1, 2 and 3. 27 manoeuvres were done during the uneventful flight.

Table 62: FT28 - Flight information.

Flight number:	28
Flight configuration:	Flutter-safe
Flight date:	11-May-2023 11:35:54
Take-off time:	11:40:44
Landing time:	12:11:33
Total flight time:	00:30:49
Total fuel used:	10.57 kg

FT29 Coordinated turn with flutter configuration, flexible mode maneuvers and flutter stopper tests

Flight test 29 was the first flight test with the flutter configuration. Steady circle flying was planned to collect structural data in-flight. Furthermore, flexible manoeuvres with flutter configuration and first test of the flutter-stopper functioning in-flight were planned.

Table 63: FT29 - Flight information.

Flight number:	29
Flight configuration:	Flutter
Flight date:	12-May-2023 06:03:50
Take-off time:	06:08:42
Landing time:	06:42:53
Total flight time:	00:34:11
Total fuel used:	9.92 kg

Very calm winds (according to the METAR just before takeoff - below $2m/s$) meant that the altitude hold and autothrottle modules could follow the commanded values very well (Figures 313 and 314). Later on this flight was considered as one of the calmest flights of the whole project.

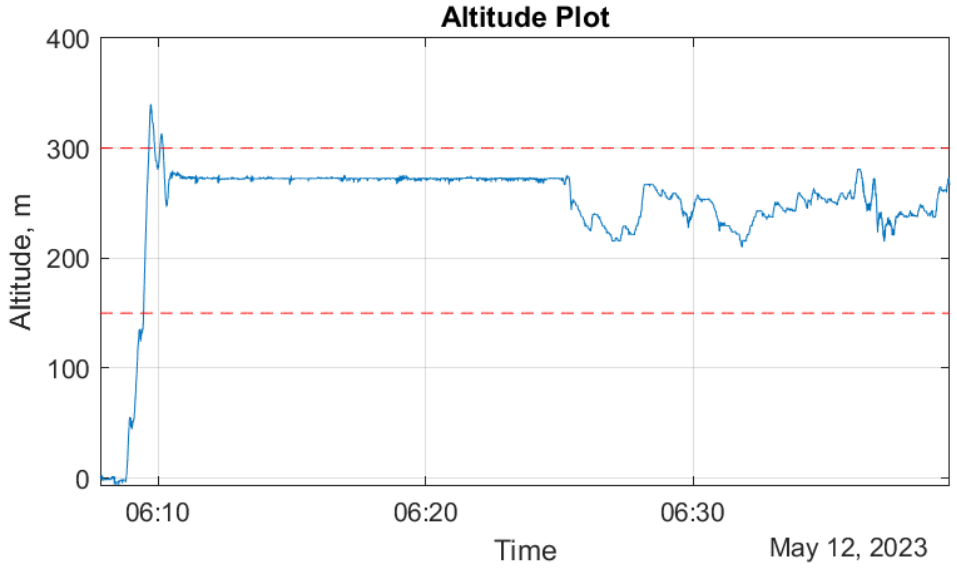


Figure 313: FT29 Altitude plot.

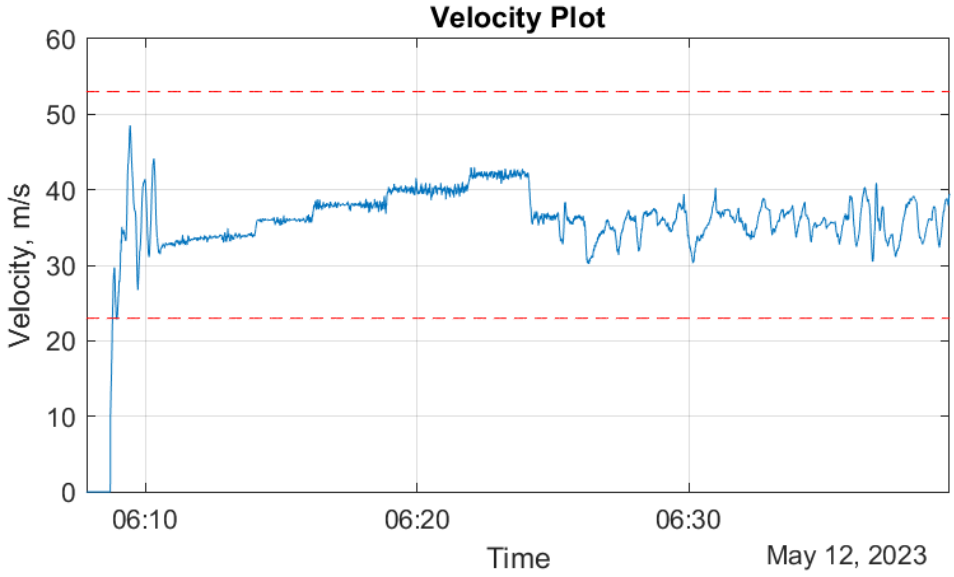


Figure 314: FT29 Airspeed plot.

During the second part of the flight, manoeuvre injections for flexible mode check were done. Then, the flutter stopper was triggered. This meant that the flutter weight at the tip of the flutter rod is shot from aft to fore position, turning the configuration flutter-safe. Even though the pilots did not notice any change on the aircraft, the accelerometer measurements on the wings could confirm that the device functioned as expected. After that, four more manoeuvres were done in order to compare the structural modes of the two configurations back-to-back.

This flight was also noted as breaking the record of how fast can the flight test crew prepare for a flight. It was 21 minutes from leaving the hangar at EDBC to being take-off ready.

FT30 Steady circles with flutter configuration and DLR controller on.

12th of May was the last day of the first week of the 4th Flight Test Campaign. After a successful Flight Test 29 in the morning it was decided that testing one of the flutter controllers in-flight would be extremely beneficial. If any problems were encountered, the next week could be used to debug the issues and prepare for flutter testing the week after. The flutter controller from DLR would be tested while flying steady circles and slowly increasing the airspeed stepwise from $36m/s$ to $42m/s$.

The timeline for preparing the flight for testing the flutter controller was very tight. At the moment when it was decided to do the flight, the software of the flutter controller was still being implemented on the aircraft and tests being done. It was decided to introduce a hard limit for the flap deflection commanded by the controller to ± 5 degrees. During the previous flight, chirp signals (frequency sweeps) were injected on the same outboard flaps with amplitude of 5 degrees. Therefore, if these limits are implemented, no structural damage was expected no matter what the controller does in-flight. Additional modifier was implemented to test the controller in both positive and negative direction, as it was not clear at that point which direction is the correct one.

First test was carried out with the controller turned on, for 2s. Nothing was noticed. Then the duration was increased to 5s. Following the wingtip Inertial Measurement Unit (IMU) accelerations in the ground station, everything seemed normal. Then we let the controller run longer, for around 10s, which also was good. The modifier was changed to negative sign and same tests were done without any visible differences, so the negative sign was left.

Table 64: FT30 - Flight information.

Flight number:	30
Flight configuration:	Flutter
Flight date:	12-May-2023 13:45:49
Take-off time:	13:50:41
Landing time:	14:18:39
Total flight time:	00:27:58
Total fuel used:	8.26 kg

Next, the steady circle test points were planned. The required modules were assigned to the two available autopilot slots as before. An airspeed command of $34m/s$ was sent to the aircraft and the pilot switched to the autopilot mode. At this point it was noticed that the airspeed did not follow the commanded value and it was slowly increasing. The Flight Test Engineer sent the command to the aircraft again without any effect. At this point it was noticed in the airspeed graph on the Engineering Data Link that the airspeed has already climbed above $40m/s$ and was quickly approaching $50m/s$. At that moment a command was issued to the pilot to take back the command from the autopilot and decrease the airspeed. Pilot switched to manual and climbed to decrease airspeed (see the spike in the airspeed, Figure 315). As no one from the Flight Test Crew expected the autothrottle mode to misbehave in this manner, it took around 3-5s to respond. It was noted that on the telemetry link the maximum recorded airspeed was above $50m/s$. Later on it was confirmed that the maximum speed at that point was actually $57.4m/s$, well above the assigned limits.

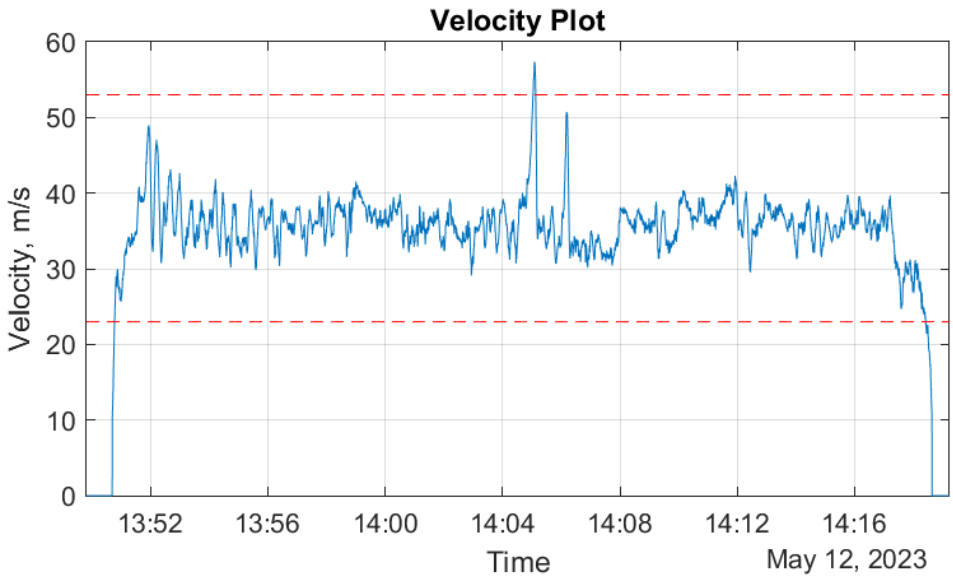


Figure 315: FT30 Airspeed plot.

A second attempt was made to repeat the same test, but when the engine was commanded full thrust the tests with the autothrottle were aborted.

It was then decided to fly the circles in manual mode at 36 and at $40m/s$. This was done with the modifier at both positive and negative values. After that the flutter stopper was triggered and landing followed.

Soon after landing it was realised that a simple programming mistake with the implementation of the

modifier had also influenced the autothrottle module, which consequently commanded the engine to go full thrust. Furthermore, after analysing the flight test data it was noticed that the flutter controller quickly saturated the preassigned maximum deflection values, so no expected dampening of wing oscillations happened in-flight.

Finally, even though the actual airspeed was above the expected flutter speed, no flutter was noticed during flight. But this incident provided a unique dataset for further flutter analysis.

Flight Test 30 concluded the first week of the last campaign with overall positive results.

FT31 Steady circles with flutter configuration and SZTAKI controller on

The flight test crew had come back to the EDBC airport on 21st of May, but due to the required preparations for flutter tests and bad weather on the 23rd of May, no flights could take place. The next flight, Flight Test 31, was done on the early morning of 24th of May.

After fixing the bugs found in the previous flight, further flutter controller tests could be carried out. At this point recalculation of the flutter speeds including the results from the ground vibration tests was done. The new airspeed, confirmed by two independent calculations was set at $56m/s$ instead of the previous value of $49 - 51m/s$. Therefore, it was required to gather more steady flight data at airspeeds higher than the $42m/s$ flown during Flight Test 29 (Section 2.3.3). The plan for the Flight Test 31 was therefore to first gather the steady data for up to $48m/s$ and then perform the controller tests at multiple airspeeds.

Table 65: FT31 - Flight information.

Flight number:	31
Flight configuration:	Flutter
Flight date:	24-May-2023 07:05:53
Take-off time:	07:10:46
Landing time:	07:42:52
Total flight time:	00:32:05
Total fuel used:	10.36 kg

The uneventful flight test went according to the plan. Steady turn data was collected (Figure 316) and functioning of the flutter controllers was confirmed. Flap oscillations were visible when the flutter controllers were turned on in-flight, and no influence on handling qualities was observed.

From the airspeed plot (Figure 316) it is noticeable that the weather was significantly gustier than during the Flight Test 29 (Figure 314) where the airspeed tracking was smoother.

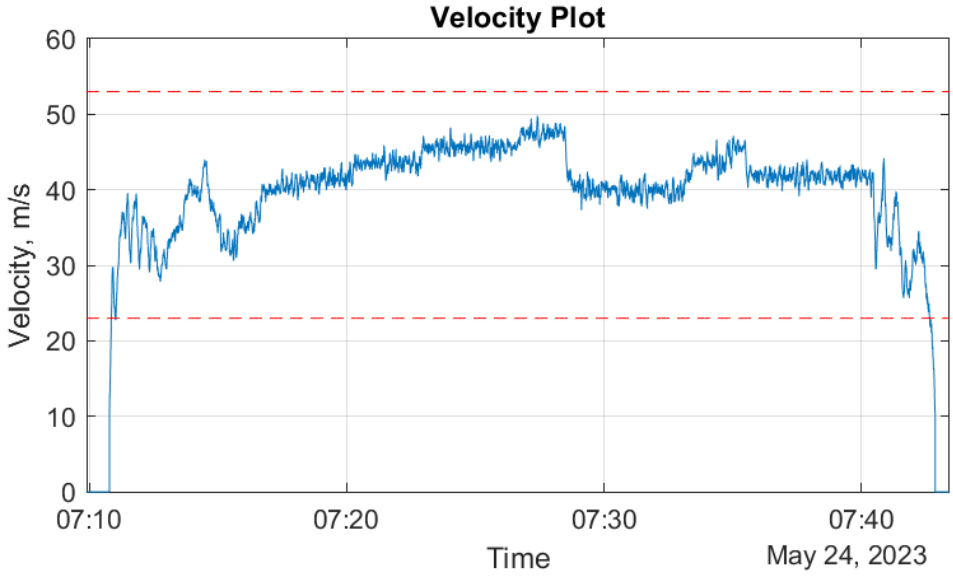


Figure 316: FT31 Airspeed plot.

FT32 - Open loop flutter speed tests

Further investigations towards the flutter speed were planned for the Flight Test 32. During the flight, 6 symmetric aileron doublets were injected (only one was successfully triggered) and steady circles up to $54 m/s$ were done (Figure 317).

Table 66: FT32 - Flight information.

Flight number:	32
Flight configuration:	Flutter
Flight date:	24-May-2023 14:24:02
Take-off time:	14:28:55
Landing time:	14:54:39
Total flight time:	00:25:43
Total fuel used:	10.49 kg

While flying at or above $48 m/s$, steady oscillations of the wing were visible not only from the wing IMU measurements, but also visually (Figure 318).

After the flight it was recognized that the planned coordinated turn with a radius of $800 m$ would be too difficult due to bad visibility and also the wind drift during a single circle. It was therefore decided to reduce the turn radius to $700 m$.

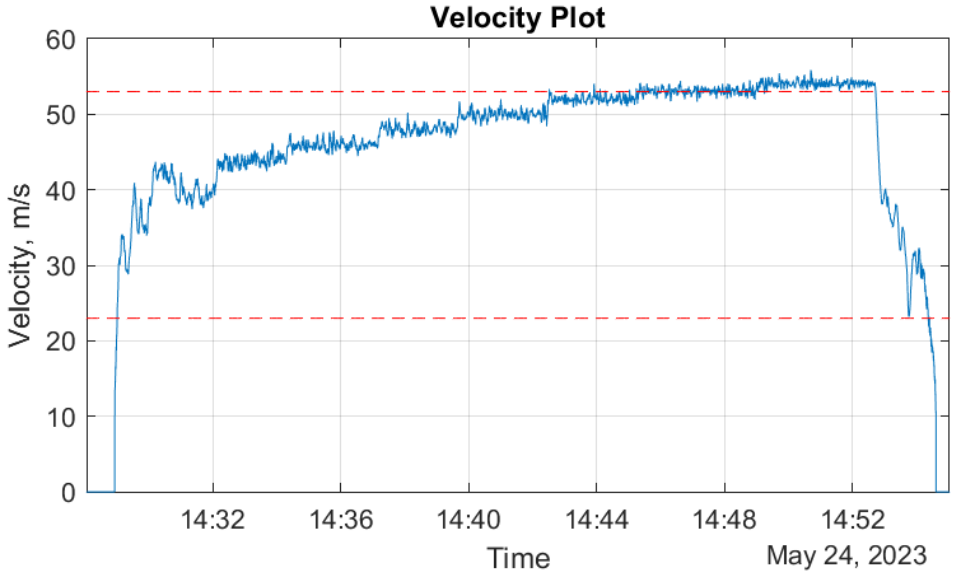


Figure 317: FT32 Airspeed plot.

Figure 318: Steady wing oscillations while flying at $54 m/s$ during the FT32. Here, two frames at maximum deflections of the same oscillation are overlayed.

FT33 - DLR Flutter Controller Tests

The next step was to further expand the flutter controller envelopes. First, the DLR flutter controller was flown. First, it was engaged only for a short period of time. In the Groud Control Station the outboard flap deflection commands were followed and if no unexpected behaviour was observed, the time could be extended. This was then done for airspeeds of 44, 48 and $50 m/s$.

After that, the controller was engaged while flying in steady circles at 44, 46 and $48 m/s$, for $90 s$ each time. Before engaging the controller, an uncontrolled segment at the trim airspeed was flown for $90 s$ as well.

Table 67: FT33 - Flight information.

Flight number:	33
Flight configuration:	Flutter
Flight date:	25-May-2023 12:19:37
Take-off time:	12:24:28
Landing time:	12:47:43
Total flight time:	00:23:14
Total fuel used:	9.19 kg

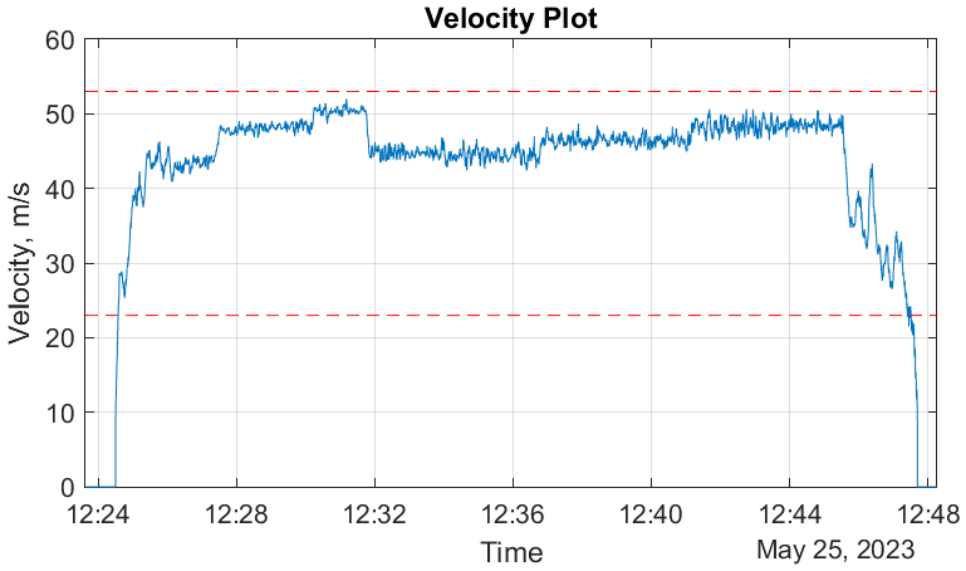


Figure 319: FT33 Airspeed plot.

FT34 - SZTAKI Flutter Controller Tests

The next flight was identical as the previous one, but this time SZTAKI flutter controllers (first and second one) were tested. After testing both for a short time at $48 m/s$, the second controller was dismissed and the circles were carried out only with one controller.

Table 68: FT34 - Flight information.

Flight number:	34
Flight configuration:	Flutter
Flight date:	25-May-2023 14:26:24
Take-off time:	14:31:12
Landing time:	14:55:32
Total flight time:	00:24:20
Total fuel used:	9.28 kg

Both controllers - DLR and SZTAKI - showed positive results for dampening the wing torsional mode.

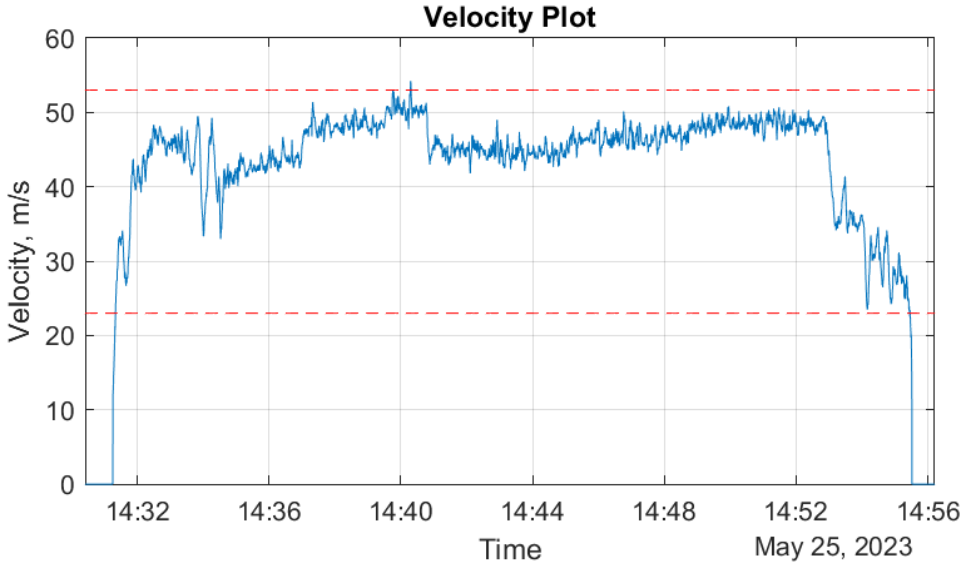


Figure 320: FT34 Airspeed plot.

FT35 - DLR Flutter Controller Tests

26th of May was the final flight day of the project. Everyone was keen to test the performance of the controllers beyond the estimated flutter speed. DLR's controller was first in line.

The plan was to start at $50m/s$ and perform the same procedure as before - fly steady circles without the controller for $70s$, then turn on the controller and continue flying with the controller for $70s$. The segments without the controller were done until $54m/s$. After that, the airspeed was increased in steps of $1m/s$, always with the active controller. After every step the ground team would evaluate the telemetry data and, provided no high oscillations are visible, would continue on with the next step.

Table 69: FT35 - Flight information.

Flight number:	35
Flight configuration:	Flutter
Flight date:	26-May-2023 06:59:34
Take-off time:	07:04:27
Landing time:	07:26:03
Total flight time:	00:21:35
Total fuel used:	9.51 kg

The tests with the active controller were successfully done up to $59m/s$. After that, the airspeed was reduced to $56m/s$ and open-loop flutter test was done in the following manner. The countdown for controller off command was started. After the controller was turned off, the countdown continued for $3s$. After that, the flutter stopper was triggered and the next second the pilot switched to augmented mode and reduced the airspeed by pulling up. Sadly, after landing it was discovered that the disengagement command for the controller was not received by the aircraft, so the test did not work as planned.

It also has to be noted that the Flight Test Crew became very good at correcting the trajectory for wind drift. When it was noticed that the coordinated turn trajectory will leave the allowed flight geography, the Flight Test Operator asked the pilot to switch to straight and level flight mode. After flying straight against the wind for a few seconds, coordinated turn autopilot mode was switched back. Good corrections

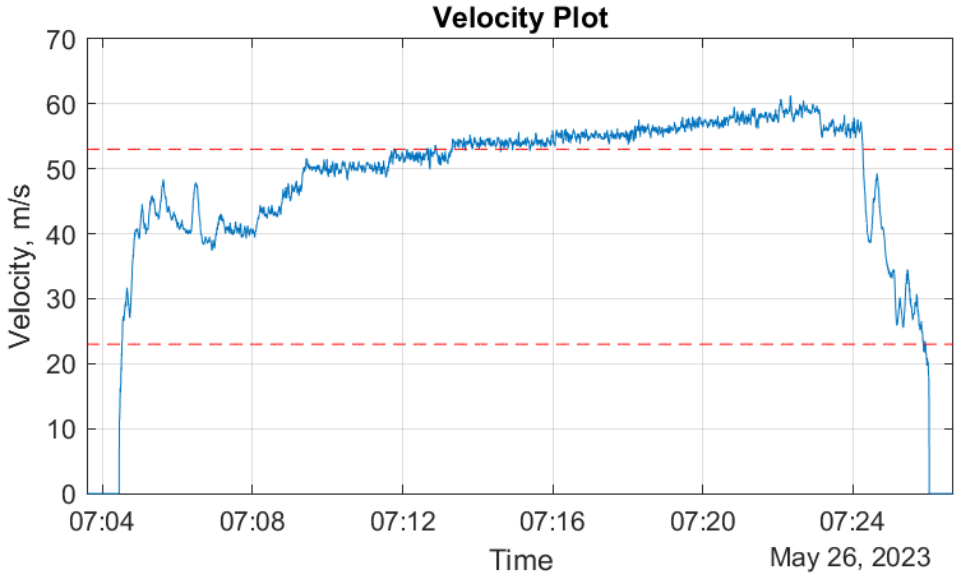


Figure 321: FT35 Airspeed plot.

resulted in a trajectory shape of a constant radius circle on eastern side and straight line on the western side of the flight trajectory (Figure 322).

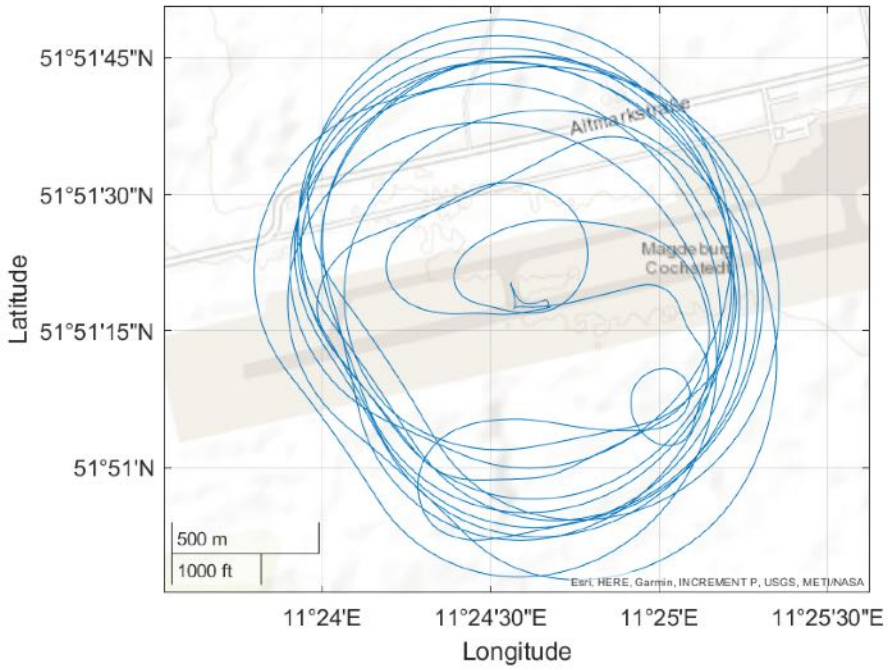


Figure 322: FT35 Trajectory plot.

FT36 - SZTAKI Flutter Controller Tests

Flight Test 36 was identical to the Flight Test 35 in terms of the flight plan. Only this time SZTAKI controller was tested instead. Once more, segments without the controller up to $54m/s$ were backed up by segments with the controller. After that the flutter controller was always on and the airspeed was stepwise increased up to $59m/s$.

Table 70: FT36 - Flight information.

Flight number:	36
Flight configuration:	Flutter
Flight date:	26-May-2023 09:45:36
Take-off time:	09:50:28
Landing time:	10:11:41
Total flight time:	00:21:13
Total fuel used:	9.42 kg

Open-loop flutter test procedure as described for the previous flight was repeated. But this time it was waited for the Flight Test Engineer to confirm that the flutter controller is turned off. Only then the countdown started for the flutter stopper to be triggered.

FT37 - Open-Loop Flutter Tests and DLR Flutter Controller Test

After seeing both of the flutter controllers active beyond the calculated flutter speed of $56m/s$ the only remaining goals of the test campaign was to confirm the open-loop flutter speed and, if possible, expand the airspeed envelope with the controller engaged as far as possible. Due to some minor differences in the software, DLR's flutter controller was chosen for the last flight of the project.

The plan for the flight was to accelerate to $54m/s$ and then keep increasing the airspeed by $1m/s$ until flutter could be observed from the available telemetry data. This was planned up to $60m/s$. After that, the airspeed would be commanded down to $56m/s$ and the flutter controller would be engaged. Finally, the airspeed would be increased to as far as the remaining fuel allows. If anything serious happened, the flutter stopper would be triggered.

Table 71: FT37 - Flight information.

Flight number:	37
Flight configuration:	Flutter
Flight date:	26-May-2023 13:41:40
Take-off time:	13:46:32
Landing time:	14:05:55
Total flight time:	00:19:22
Total fuel used:	9.42 kg

The flight took off in turbulent conditions with wind speeds of around $5m/s$. After the climb to $260m$, the airspeed command of $54m/s$ was sent to the aircraft. Nothing unexpected was noticed from the ground, so the airspeed was stepwise increased all the way up to $59m/s$ (Figure 323). After this section of tests the speed was reduced to $56m/s$, the flutter controller was turned on and the aircraft was again accelerated up to $61m/s$.

At some point during the maximum speed segment it was noticed that the right flutter rod is hanging on the wing in an unfamiliar configuration and that the left one is missing altogether (Figure 324). After this was noticed, further tests were cancelled and the pilots went for landing.

A detailed investigation of the flight test followed. It was necessary to figure out what exactly happened, to be able to tell if the rods broke off due to flutter condition or if there was structural damage experi-

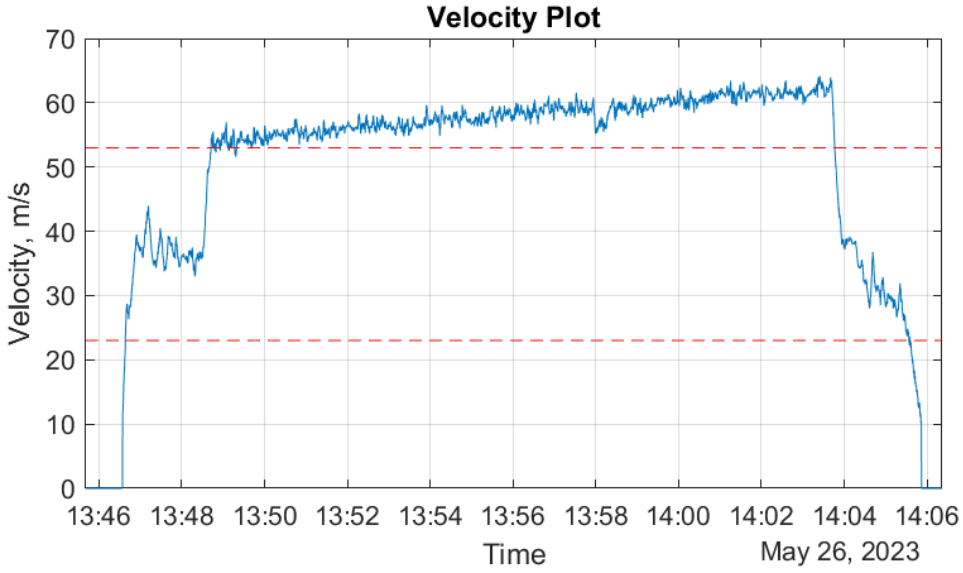


Figure 323: FT37 Airspeed plot.

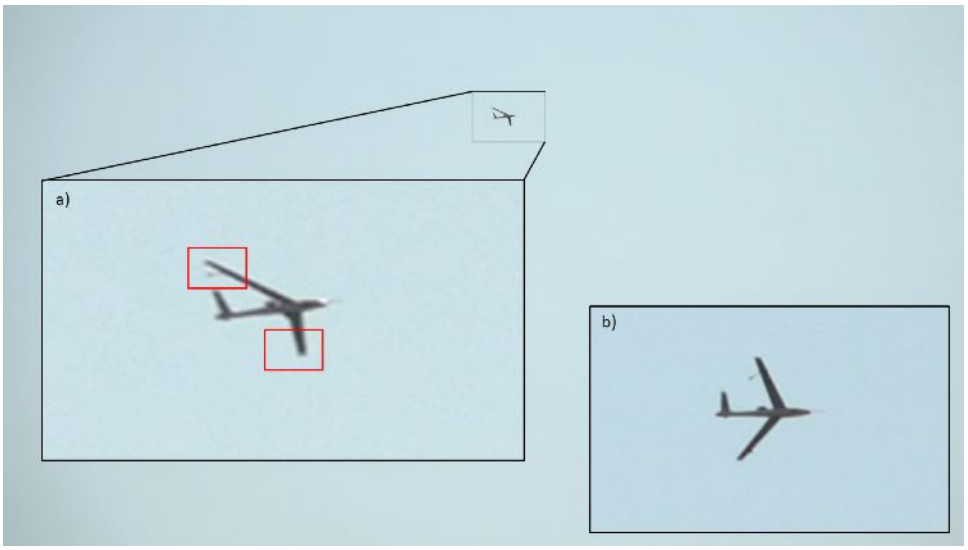


Figure 324: The view of the flying aircraft from the camera tracker during FT37: a) Left flutter rod missing after 6 minutes of flight; b) Right flutter rod hanging from the wing.

enced before that. After analysing the available data, the important events were noted in Table 72. Data from the first minute after accelerating to $54 m/s$ is also plotted in Figure 325. The 16 seconds where biggest vibrations of the left wing were experienced are also plotted in Figure 326. In the plots, IMU4 is the trailing edge IMU unit in the left wingtip and IMU10 is the trailing edge IMU unit in the right wingtip.

Note that the following things were different on this flight when compared to the flights before:

- The air was very turbulent, which could result in higher excitation energy from turbulence.
- The left flutter rod assembly was exchanged before the flight because of a stuck flutter stopper

Table 72: Table of events during the Flight Test 37. Tail left and right denotes the timestamps visible in the videos.

	Tail left	Tail right	FCC log	Event
0:00:00	14:52:38	15:45:40		Engine ON
0:00:21	14:52:58	15:45:59		Control check finished
0:00:29	14:53:07	15:46:08	13:46:00	Engine starts to spool up
0:01:06	14:53:44	15:46:45	13:46:37	Lift off
0:01:33	14:54:10	15:47:12	13:47:04	$38-40m/s$ reached and stable
0:03:00	14:55:37	15:48:39	13:48:31	Acceleration started.
0:03:11	14:55:48	15:48:50	13:48:42	Some vibrations visible on the left flutter rod, but nothing seems to be broken.
0:03:14	14:55:51	15:48:53	13:48:45	54m/s reached and stable
0:03:22	14:55:59	15:49:01	13:48:53	Maybe some free-play is visible on the left flutter-rod mount. At that point the airspeed varies in between 53.5 and 54.8m/s.
0:03:27	14:56:04	15:49:06	13:48:58	Vibrations on the left rod. Airspeed varies from 53.2 to $55.4m/s$
0:03:33	14:56:10	15:49:12	13:49:04	$57.2m/s$ point. Unsteady vibrations visible on both sides.
0:03:46	14:56:23	15:49:25	13:49:17	First big vibration. Left flutter rod is broken at the mounting point. Oscillations, but no damage is visible on the right side.
0:03:54	14:56:31	15:49:33	13:49:25	Second big vibration. Left flutter rod is broken even more. Oscillations, but no damage is visible on the right side.
0:05:13		15:50:52	13:50:44	$57.6m/s$. Constant oscillation of the right side.
0:06:44	14:59:21	15:52:23	13:52:15	Left flutter rod breaks off, hits the flap and flies away. Big vibrations on the right side, here the right side gets broken as well.
0:07:47	15:00:24	15:53:26	13:53:18	Right side is visibly broken.
0:08:15	15:00:52	15:53:54	13:53:46	Right shaken a lot.
0:11:55	15:04:32	15:57:34	13:57:26	Right side is hanging on the wire/small string of carbon.

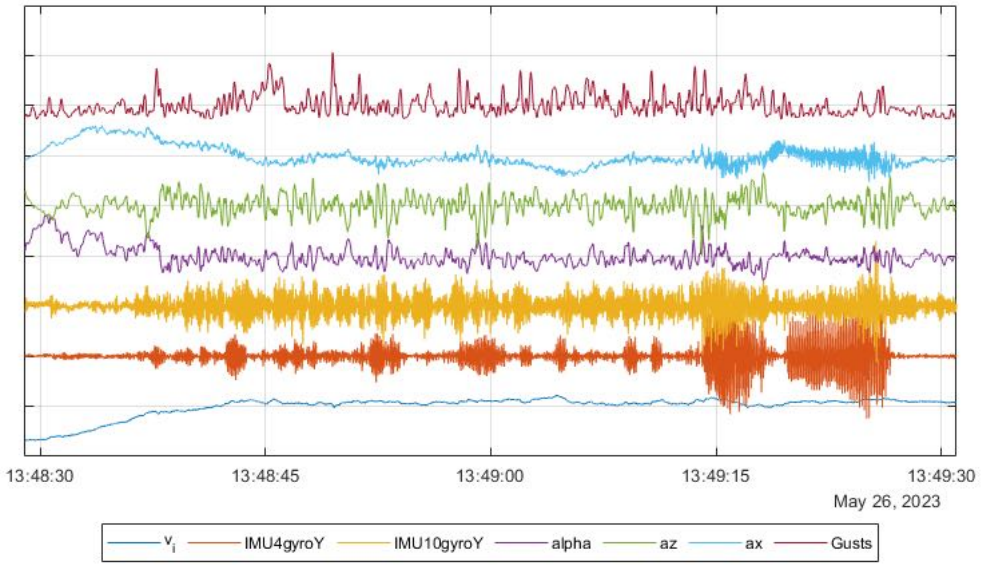


Figure 325: Data of the first minute when accelerated to $54m/s$. All variables here are normalized and offset vertically to each other for better visibility.

release mechanism. The exchanged assembly was identical to the previous one.

Record three flights were done on the day, successfully ending the last Flight Test Campaign of the

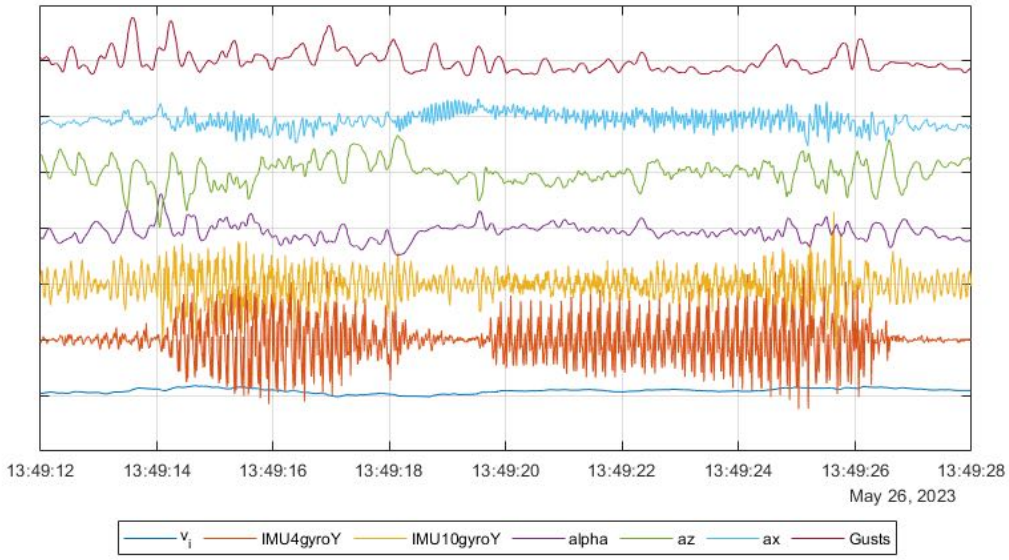


Figure 326: Data of the 16 seconds when highest oscillations of the left wing are noted. All variables here are normalized and offset vertically to each other for better visibility.

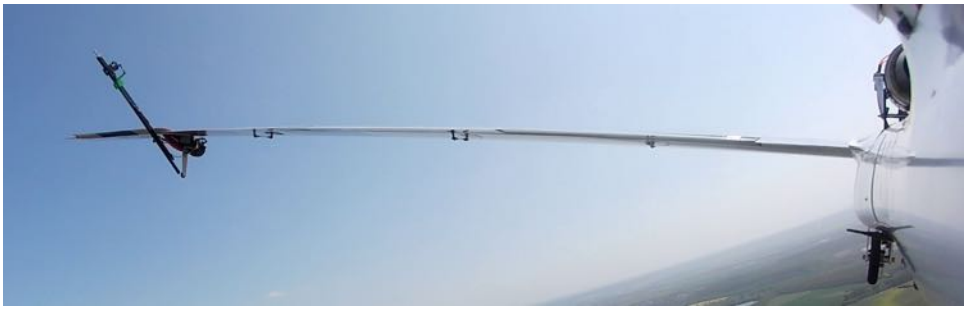


Figure 327: The moment just before the flutter rod completely breaks off during the FT37.

project FLIPASED.

Summary of the 3rd and 4th Flight Test Campaign During the two flight test campaigns in the last year of the project, 14 test flights were done. Due to the increased fuel tank capacity and lack of system issues during flight, the average flight time was 27 minutes (compared to 16 minutes in 2022). In total, more than 380 minutes of flight time were collected.

The flight tests successfully covered most of the goals raised after the crash of the demonstrator in 2022. A wide range of data was collected for model updating, from steady to unsteady test points. The autopilot modes were updated and prepared for flutter tests. Flutter test procedures were streamlined. Finally, flutter controller tests were performed at both sub-critical and critical airspeeds. Both controllers (SZTAKI and DLR) were proven to reduce the oscillations of the targeted modes.

Flight Test Data The data set generated within the project is published according to the standards of the Open Research Data Pilot, i.e. as a type of open data which is focused on publishing observations and results of scientific activities available for anyone to analyze and reuse. In the project, the various types of shared data include:

- Time series data - of flight tests in .mat files
- Flight test reports
- Flight test cards describing the goals, parameters and important events of the test

The corresponding flight control algorithms within Matlab/Simulink environment are published as well for reference. They can be found at:

Flutter control design and analysis scipts.

Note that the scripts are optimized for Matlab r2016b version, and often yield worst results with newer versions of Matlab. The scripts are using sub-modules from [41].

All data are published and uniquely identified through the CONCORDA (https://science-data.hu) scientific data repository system.

Flight test data structure

The data is uploaded in a single *.rar or *.zip archive file which contains:

- 01_Processed_Data
  - xy_post.mat
  - xy_TMS.mat
  - xy_jeti_log.mat
- 02_Analysis
  - xy_analysis_file.mat
  - xy_jeti_log_alligned.mat
  - FlightTestReport.docx
  - xy_flight_phases.xls
- Flight test card
- xy_extracted_maneuvers.xls

The flight test card describes the flight test sequence and the notes about the flight itself. extracted_maneuvers.xls contains the description of the different parts of the flight itseld with basic statistics and the start and end time of each test segment.

The xy_analysis_file.mat contains 247 UTC time stamped variables, including basic analysis variables about the flight itself.

The xy_post.mat file contains post-processed data about the flight itself, post processing means proper time stamping and outlier removal from the data, as well as adding calibration values to certain measurements where _raw logs are not in SI units.

There are also files named _TMS.mat related to the thrust measurement unit and _jeti_log.mat files related to the inputs and measurements received from the RC system.

For better understanding and to reduce the learning curve a sample flight test data analysis script for FT34 is also available at FT34 at CONCORDA. This includes all the lever arms and dimensions as well as the basic analysis scripts required to get familiar with the data.

Table 73: Flight test data overview

Flight test	Test Objective	Airfield	Link
FT 7	Augmented mode and auto-throttle mode tests. Familiarization for the pilots. Test trim points with different flap settings.	EDMO	link
FT 8	Throttle injection tests. Familiarization for the pilots.	EDMO	link
FT 9	Throttle injection tests. Familiarization for the pilots.	EDMO	link
FT 10	Augmented mode check, pilot training.	EDBC	link
FT 11	Augmented mode check, pilot training.	EDBC	link
FT 12	Autothrottle check.	EDBC	link
FT 13	Course angle and horse race pattern.	EDBC	link
FT 14	Course angle and horse race pattern.	EDBC	link
FT 15	Autothrottle envelope check.	EDBC	link
FT 16	Autothrottle envelope check.	EDBC	link
FT 17	Rigid body mode maneuvers.	EDBC	link
FT 18	Pilot Training	EDBC	link
FT 19	Autopilot mode checks.	EDBC	link
FT 20	Pushover pullups and engine effects.	EDBC	link
FT 21	Rigid body mode maneuvers.	EDBC	link
FT 22	Rigid body mode maneuvers.	EDBC	link
FT 23	Rigid body mode maneuvers.	EDBC	link
FT 24	First flight of P-FLEX and the -1 wing, pilot training, autopilot checks.	EDBC	link
FT 25	Autothrottle, course angle hold test, manoeuvre injections.	EDBC	link
FT 26	Further checks on auto-throttle controller and the performance of it with coordinated turn. Do the manoeuvres from block 1.	EDBC	link
FT 27	Check the course angle hold, auto-throttle check through the full envelope, manoeuvres from block 1 and 2.	EDBC	link
FT 28	Manoeuvres from block 1, 2 and 3.	EDBC	link
FT 29	Steady circles with flutter configuration. Flexible manoeuvres with flutter configuration. Flutter stopper check.	EDBC	link
FT 30	Steady circles with flutter configuration and DLR controller on.	EDBC	link
FT 31	Steady circles, steady circles with flutter configuration and SZTAKI controller on.	EDBC	link
FT 32	Signal injection with flutter configuration.	EDBC	link
FT 33	Steady circles with flutter configuration DLR controller on (up to 50 m/s).	EDBC	link
FT 34	Steady circles with flutter configuration SZTAKI 4 input and SZTAKI 6 input controllers.	EDBC	link
FT 35	Steady circles with flutter configuration DLR controller on, up to 59m/s.	EDBC	link
FT 36	Steady circles with flutter configuration SZTAKI controller on, up to 59m/s.	EDBC	link
FT 37	Confirm open-loop flutter speed with increasing velocity until failure.	EDBC	link

Flight Test Data Analysis Tools The updated data processing workflow is visualized in Figure 328. In comparison to the workflow described in D3.2 Flight Test Report – Flight Test Phase #1, the following upgrades were made:

- To easier extract the segments of interest, a Graphical User Interface (GUI) was created;
- An automatic segment extraction for airbrake inputs, autopilot segments, steady-level-flight segments and turns was added;
- Thrust measurement data alignment was updated. As a result, the log delay in between the applied thrust and actual thrust was reduced.

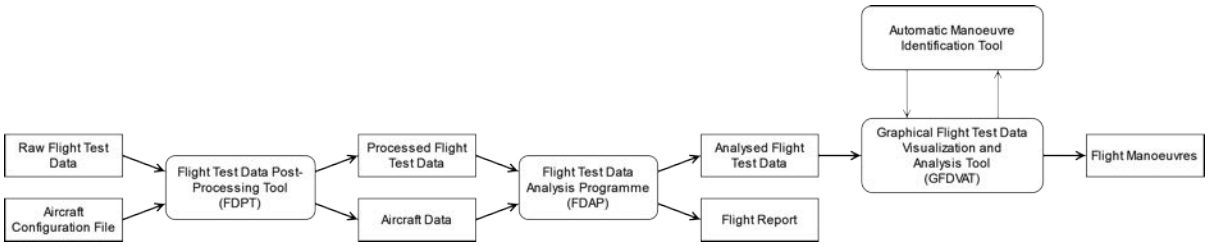


Figure 328: Flight test data processing and analysis workflow.

The created GUI allows the user to import and visualize any flight. The GUI displays the map area and the 3D visualization of the aircraft for the selected flight segment (Figure 329). Below that any data can be plotted for the selected segment and overall altitude or airspeed graph allows the user to orient better which part of the flight the current segment is from.

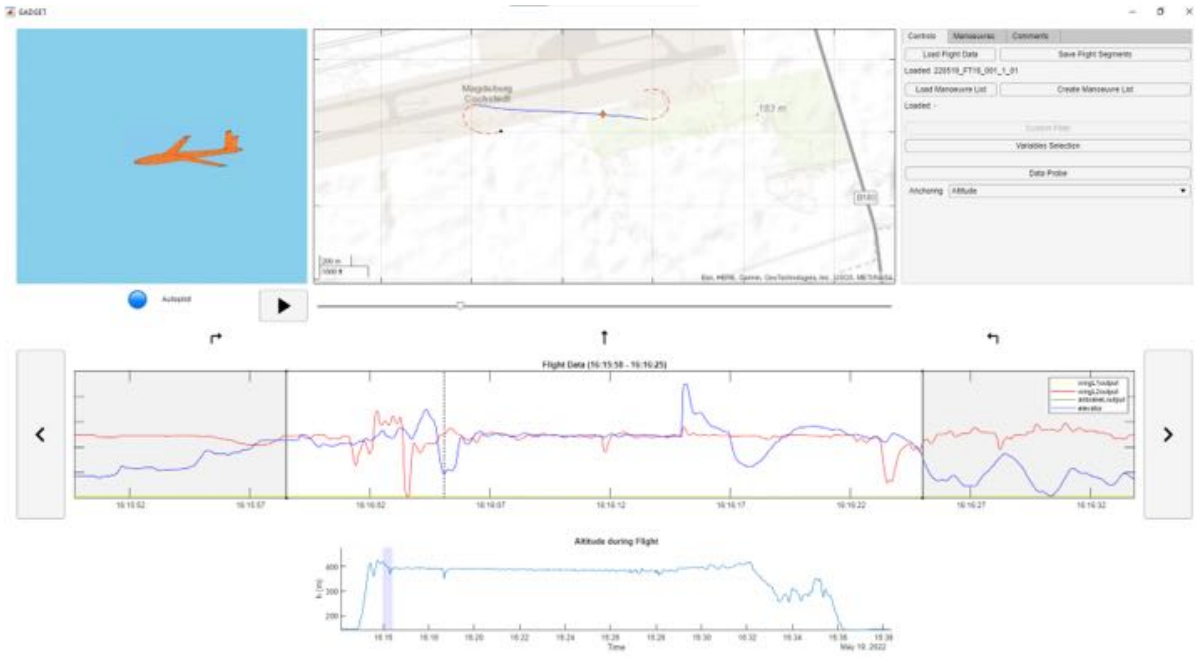


Figure 329: The Graphical User Interface for flight test data analysis.

A manoeuvre identification module is available (Figure 330). Note that in this case identification means finding the right time period where a specific manoeuvre was performed. Therefore, this module allows

the user to automatically locate all the time segments where, for example, steady-level flight was done or where the autopilot mode was changed.

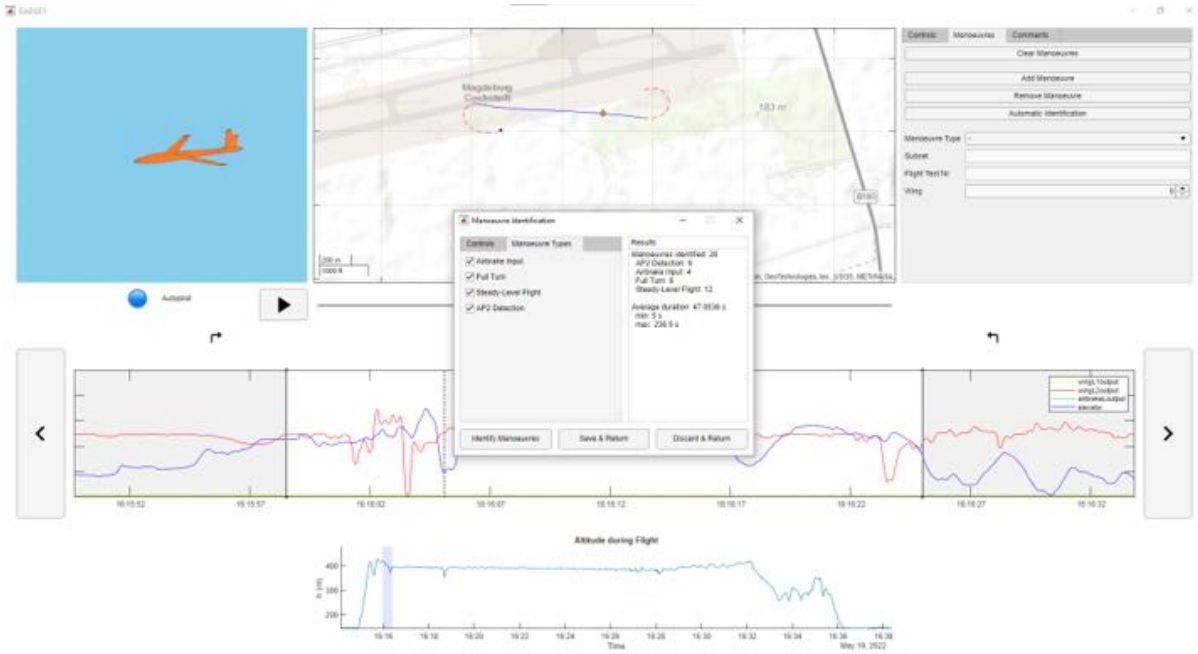


Figure 330: The manoeuvre identification module for flight test data analysis tool.

After all the manoeuvres were located in time, titles for the manoeuvres can be changed and comments added (Figure 331). The manoeuvre time segments are also shown in the graph.

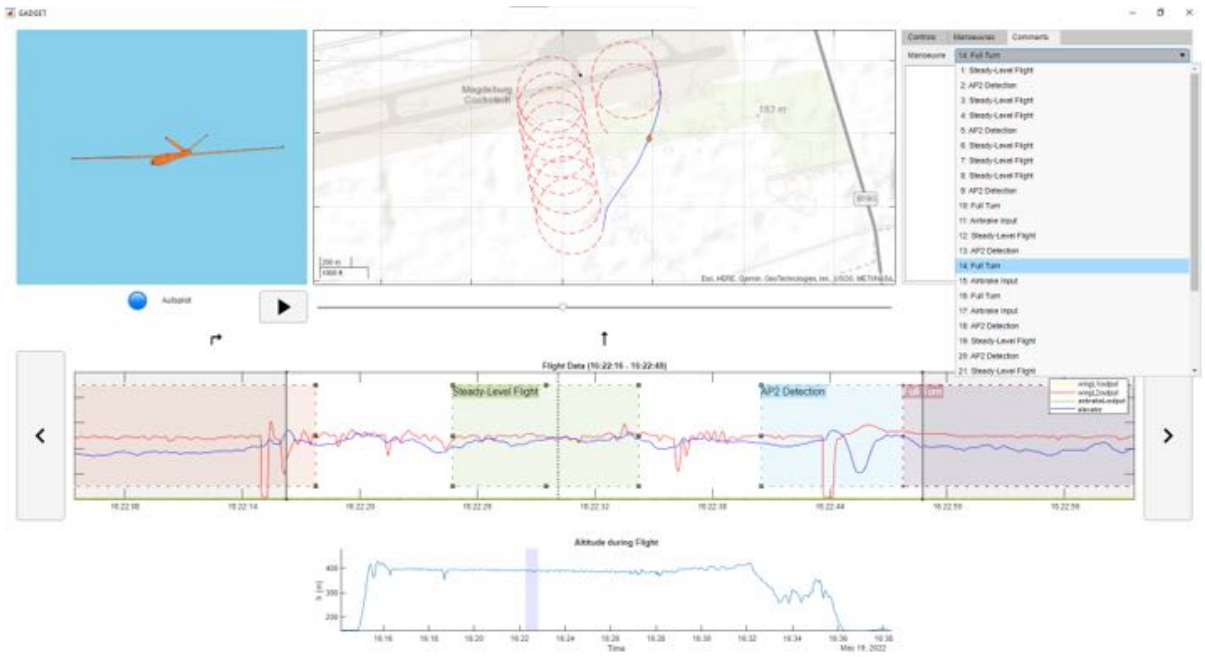


Figure 331: The identified manoeuvre time segments are displayed in the flight test data analysis tool.

Finally the manoeuvres are saved in a separate table with timestamps, titles, comments and additional data. This is then used as a basis of extracting only the required time segments from the overall test data.

Autopilot mode evaluation (SZTAKI) Various functionalities of the baseline controller have been tested during the 1st Flight Test Campaign. Among the conducted flight tests FT10, FT12, FT14 and FT16 have been successful in terms of the baseline control testing, other flight tests either focused on different aspects or had technical problems, hindering the testing of the baseline functionalities. The tested functionalities are summarized in Table 74, while each mode is evaluated separately in the forthcoming subsections. Note that in FT10 only 30 seconds of baseline controller tests were performed and hence it is not included in the analysis. Accordingly, FT12, FT14 and FT16 form the basis of the numerical evaluation of the baseline controller's performance.

	FT10	FT12	FT14	FT16
Roll attitude	✓	✓	✓	✓
Pitch attitude	✓	✓	✓	✓
Sideslip control	✓	✓	✓	✓
Altitude control		✓	✓	✓
Lateral directional			✓	
Autothrottle		✓	✓	✓

Table 74: Overview of baseline controller tests during the 1st Flight Test Campaign

During the 4th Flight Test Campaign the altitude hold and the autothrottle functionalities of the baseline controller were investigated only. Based on the results of the previous flight tests a few modifications have been carried out in order to fix the minor issues in the autothrottle loop. First, the saturation limits have been corrected, second a smoothing filter has been designed and implemented in order to eliminate the high amplitude oscillations in the commanded RPM signal. Lastly, a new gain set was proposed by based on a new engine model obtained from the in-flight thrust measurements. Note also that both control loops are acting on the longitudinal axis of the aircraft, therefore their performances are also coupled.

Roll attitude

The roll attitude loop is the inner loop controller of the lateral-directional control of the Unmanned Aerial Vehicle (UAV) designed for tracking reference bank angle (ϕ_{ref}) coming either directly from the radio control (RC), commanded by the pilot, or from the outer loop control for achieving path following through course angle. The augmented mode behavior has been thoroughly tested and verified during the previous Flight Test Campaigns, however in FT12 it has been further tested. Figure 332 shows the typical tracking performance of the roll attitude loop, where the commanded signals were generated by the pilot: fast response time and precise tracking characterize this loop. Similar performance was observed in the previous flight tests as well as in FT16 of the 1st Flight Test Campaign (FTC). In FT14, the corresponding lateral outer loop controller was also engaged, i.e. the lateral directional controller provided the reference roll angle, which can be observed on Figure 333.

	FT12	FT14	FT16	Σ
Mean error	-0.4263	-0.0649	0.0014	0.1447
RMS error	3.8970	4.5740	3.9373	4.1363

Table 75: Performance of the roll attitude control loop during the 1st Flight Test Campaign

Table 75 summarizes the numerical evaluation of the roll attitude control loop, where the mean error

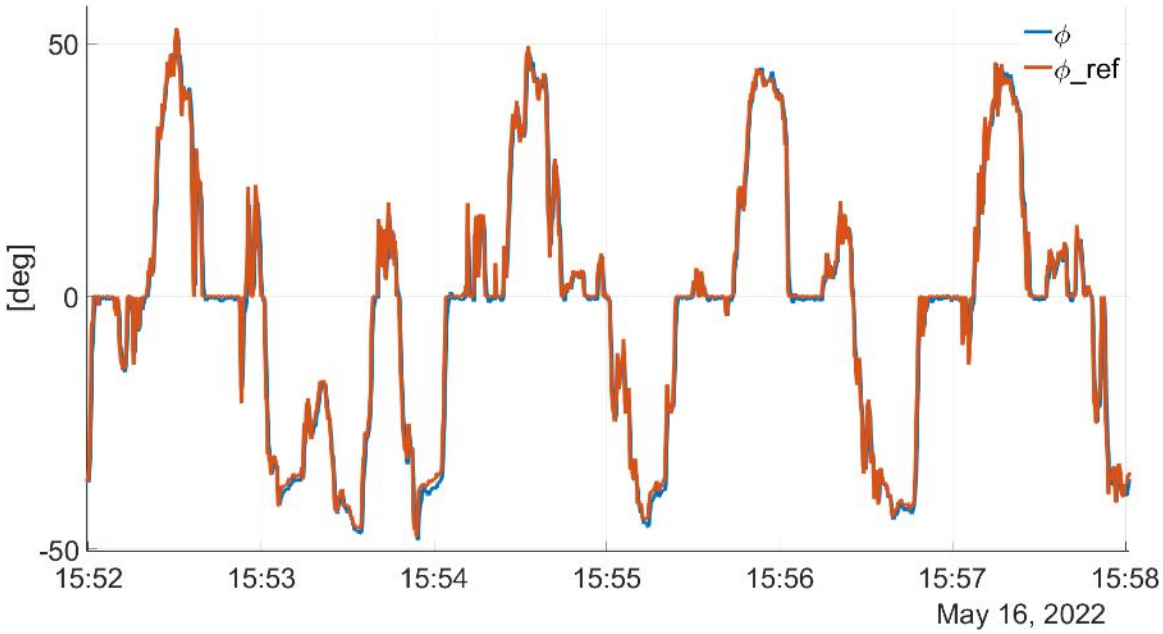


Figure 332: Roll attitude control in augmented mode during FT12

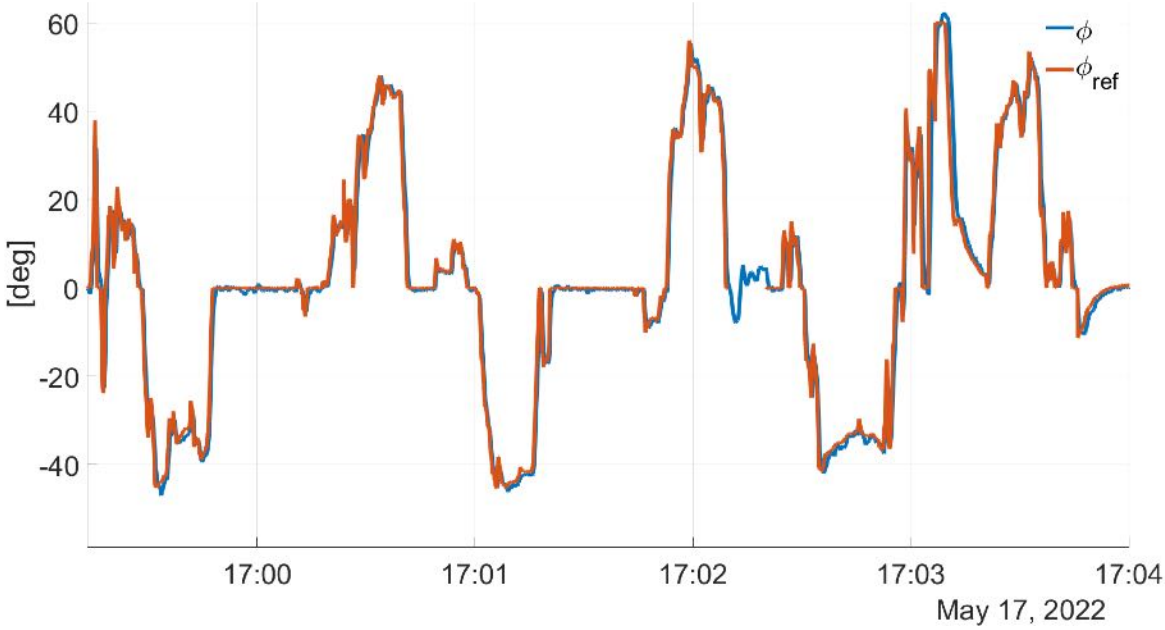


Figure 333: Roll attitude control with lateral-directional outerloop, during FT14

and the root mean square (RMS) error metrics were chosen. The overall tracking performance is characterized by a less than 0.2 degrees error, while the RMS value was 4.16 over the three flight tests. In conclusion: the roll attitude control loop provided a very satisfactory, quick tracking of the reference

signal in both augmented and fully automated modes. Accordingly the loop has passed the flight tests and no further modification was necessary.

Pitch attitude

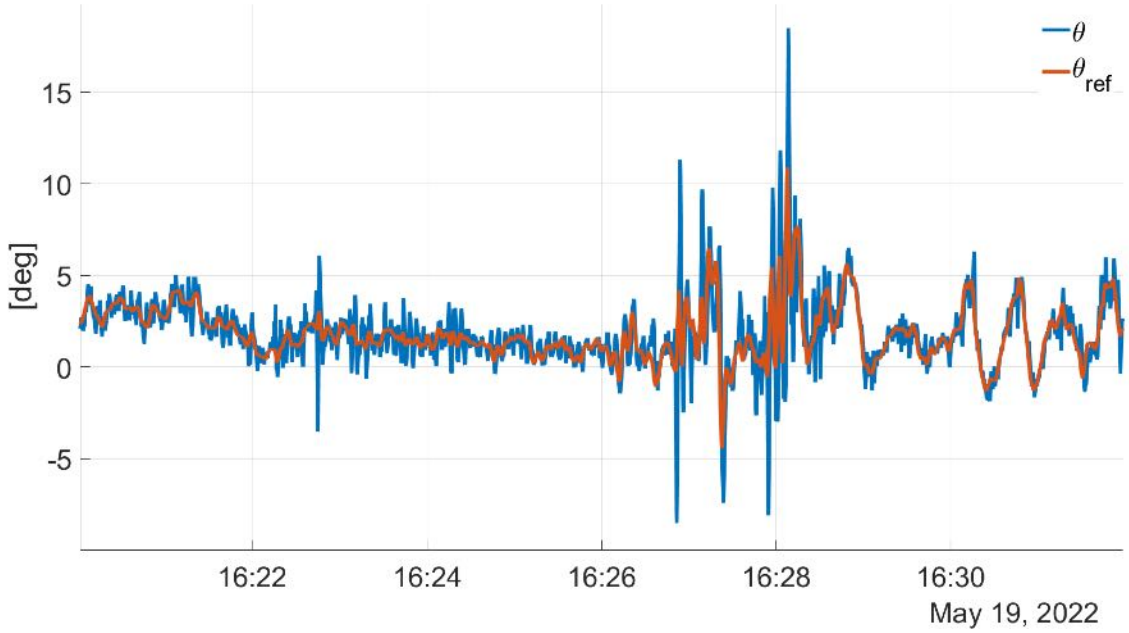


Figure 334: Pitch attitude control with altitude hold outerloop, during FT16

The pitch attitude controller belongs to the longitudinal control architecture and similarly could be operated either in augmented or in fully automated modes. The augmented mode has been tested in the previous Flight Test Campaigns, therefore only the automated functionalities have been tested in the 1st FTC this time. During these flight tests, the outer longitudinal loop, i.e. the altitude control, was engaged and provided the reference signal for the pitch attitude controller. A typical flight measurement can be seen in Figure 334, while the numerical evaluation in terms of mean and RMS errors is given in Table 76. In Figure 334 a slight oscillation can be observed, this is due to the tuning of the pitch attitude controller, where the design criteria was a slightly under-damped response with damping ratio 0.6, helping the pilots in the augmented mode.

	FT12	FT14	FT16	Σ
Mean error	-0.3687	-0.0226	-0.0940	0.1618
RMS error	4.6637	1.4618	2.0634	2.7296

Table 76: Performance of the pitch attitude control loop during the 1st Flight Test Campaign

For the pitch attitude control a similar performance was achieved as for the roll attitude loop: the mean error was less than 0.2 degrees, while the RMS error was 2.73 for the three flight tests in the 1st FTC. Accordingly, the loop successfully passed the flight tests and no further tuning was required.

Sideslip Control

The sideslip control loop is a single decoupled loop acting on the lateral dynamics of the airplane aiming to ensure zero sideslip angle for the UAV. This functionality of the baseline controller was designed to

reject wind disturbances and was always active, when the baseline controller was engaged. Accordingly it has been tested during all flight tests. Figure 335 shows the recorded sideslip values during FT16, where the autopilot was engaged for a longer time period. It can be clearly seen that both the mean and the oscillation of the sideslip angle have been decreased as compared to the uncontrolled case, where no reference command was issued (after 16:32 in Fig 335. Similar behavior was found in the other flight tests as well. The overall numerical evaluation is given in Table 77, where the uncontrolled values are also given for better understanding the effect of the sideslip control. The average tracking error is 0.015 degrees over the three flight tests (compared to the 0.75 degrees in the uncontrolled case) during the 1st Flight Test Campaign, with RMS value of 1.66 (in contrast with the 3.86 for the uncontrolled case). Note also that under specific maneuvers the sideslip angle slightly increased, but within an acceptable region.

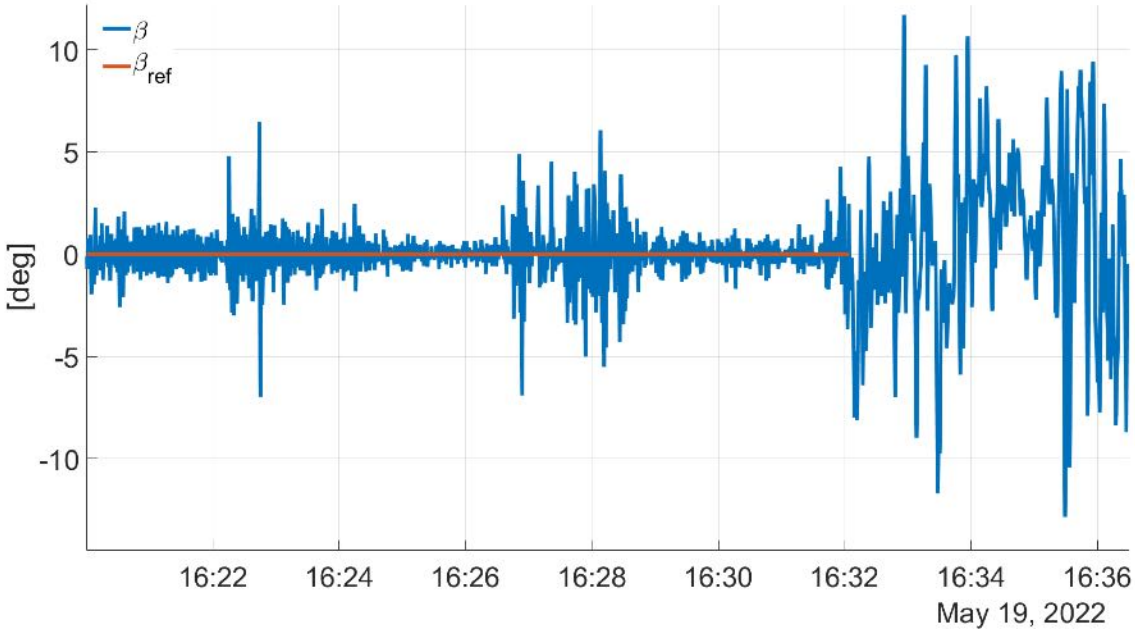


Figure 335: Sideslip control during FT16

	FT12	FT14	FT16	Σ
Error (uncontrolled)	0.0024	-1.8954	0.3534	0.7504
Error (controlled)	-0.0181	-0.0174	0.0085	0.0147
RMS (uncontrolled)	3.3511	4.0527	4.1655	3.8564
RMS (controlled)	1.6704	1.7062	1.5955	1.6574

Table 77: Performance of the sideslip control loop during the 1st Flight Test Campaign

Accordingly, the sideslip loop was considered satisfactory and required no further modificiation for the future flight tests.

Altitude control

The altitude controller is part of the longitudinal control loops and designed for keeping constant barometric altitude of the aircraft. When engaged, the controller provides reference pitch angle for the corresponding longitudinal inner loop (pich attitude controller). The main objective of the control loop

is to hold a specific constant altitude during the flight tests. For this, please see Figure 336, where the controller was active for a longer period of time. Here it can be seen that the integral action of the controller was not satisfactory: the error was decreasing slowly. This problem is also clear from the numerical evaluation in Table 78, where the mean error is about 4 meters, paired with a relatively small RMS value. Since this issue was not critical for the flight test campaigns, the performance and the controller was accepted.

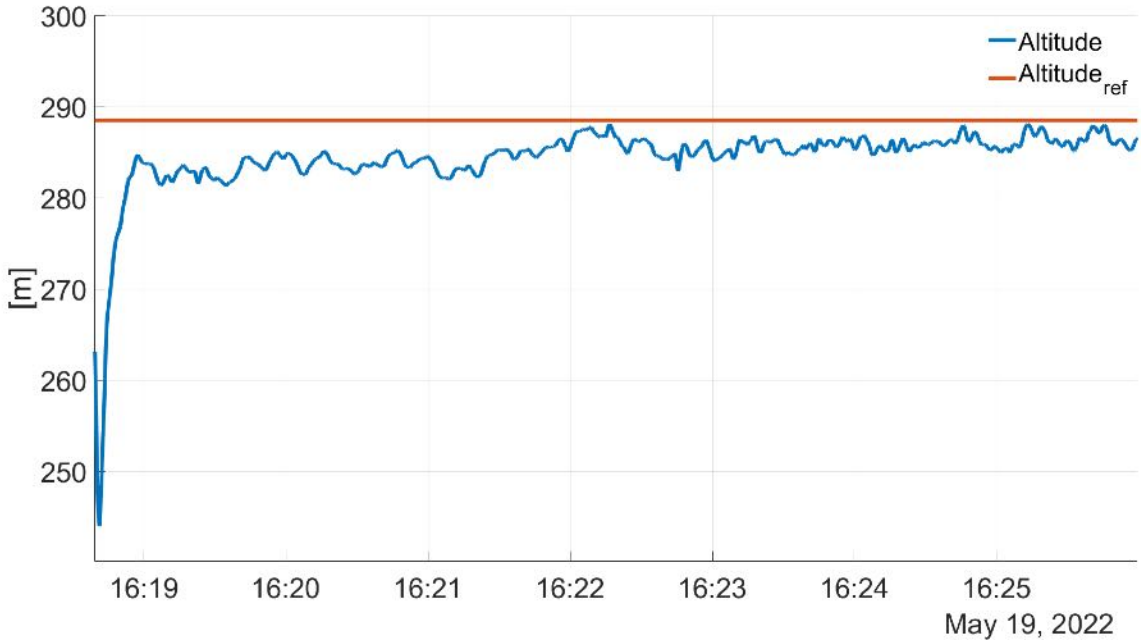


Figure 336: Altitude hold during FT16

Although it was not part of the design requirements, a step tracking test was also performed during FT12, which is depicted in Figure 337.

	FT12	FT14	FT16	Σ
Mean error	-4.0625	-4.3652	-4.2850	4.2376
RMS error	8.0787	7.0009	5.888	6.9892

Table 78: Performance of the altitude control loop during the 1st Flight Test Campaign

The altitude controller has been tested over the entire 4th Flight Test Campaign and its performance is summarized in Table 79. Some fluctuations can be observed, which are the natural consequence of the corresponding weather conditions. However, the overall absolute mean error was only 1.5 meter, in addition the previously observed large transients upon the engagement of the altitude controller has been successfully eliminated by fixing the bumpless transfer logic. This can be also seen on the time-domain Figures 338-341. During FT26, three different autothrottle controller configurations were tested, without diminishing the altitude hold capabilities, as depicted on Figure 338. Subsequent flight tests were focusing more intensely on the open and closed loop flutter tests. FT29 were the first flutter configuration flight test, while the first open loop flutter tests were conducted during FT32 (the recorded altitude hold performance is given in Figure 339). FT35 and FT36 were closed-loop flutter flight tests, where in addition to the autopilot control, DLR and SZTAKI flutter controllers were also actively controlling the aircraft's ailerons. Both the numerical (Table 79 and time domain (Figure

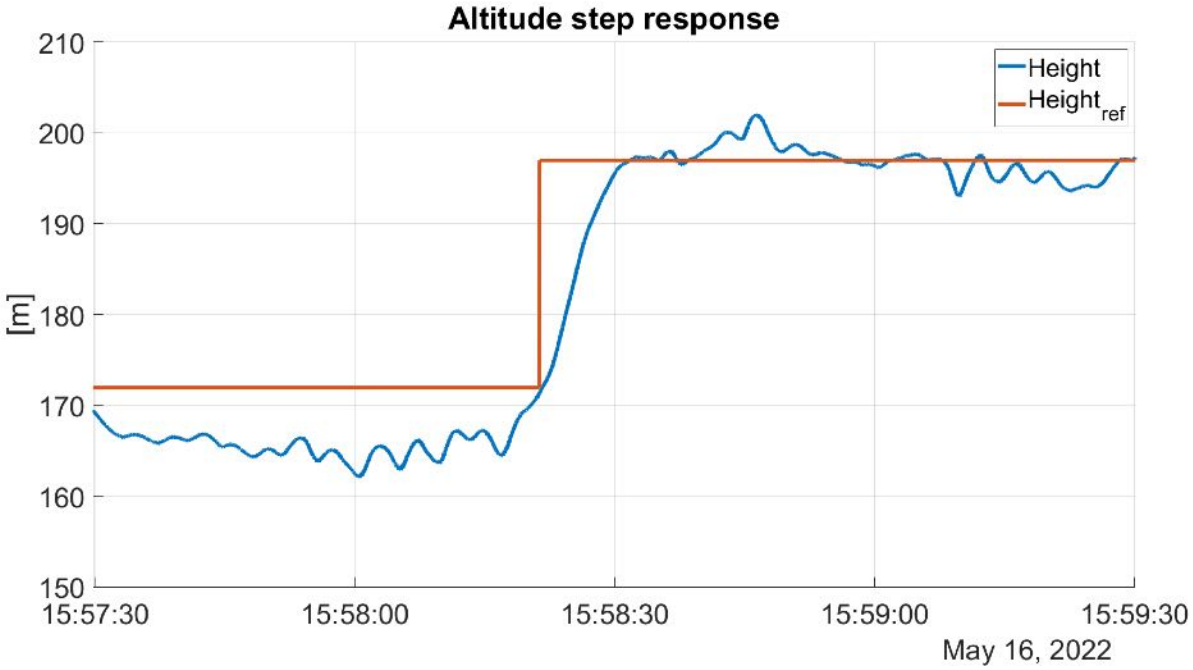


Figure 337: Altitude step reference tracking during FT12

340-341) shows that no performance degradation occurred during these flights.

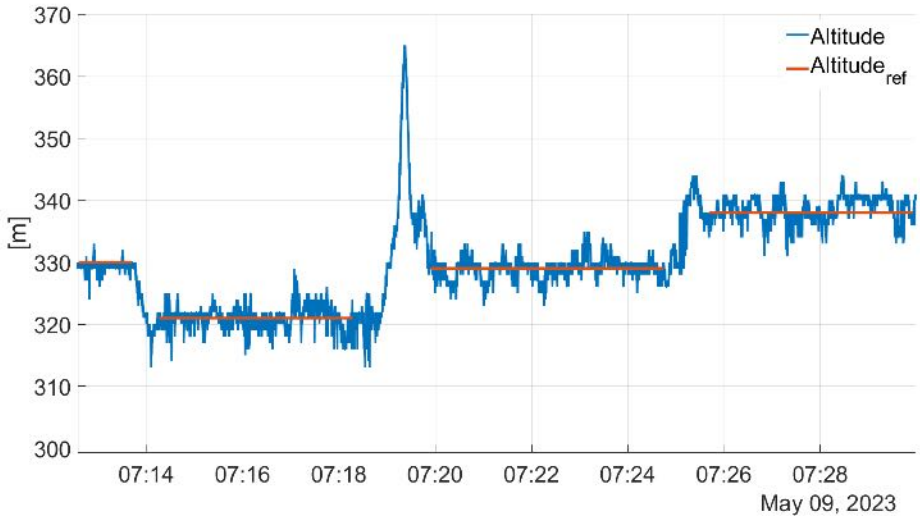


Figure 338: Altitude hold during FT26

Lateral directional control

The lateral directional control loop was responsible for tracking the course angle (flight path) of the aircraft. Based on the navigation signals a reference χ angle was computed as an input for this controller, which is then forwarded to the roll attitude loop as a bank angle command. This functionality of the baseline controller was only tested in FT14, where the corresponding tracking behavior can be seen in

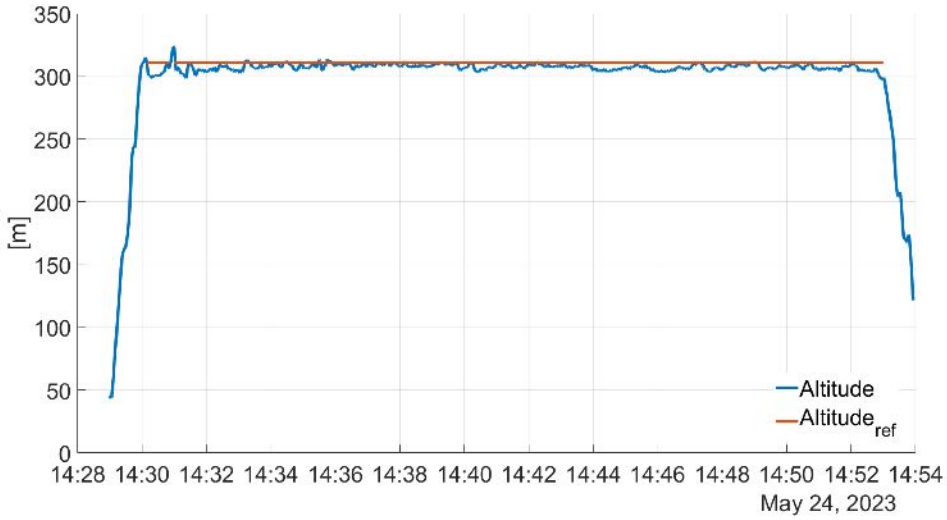


Figure 339: Altitude hold during FT32

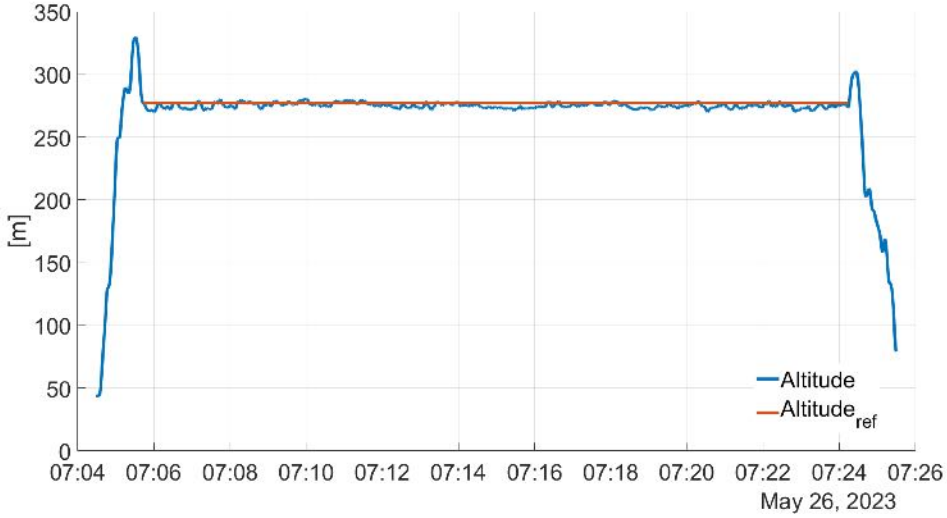


Figure 340: Altitude hold during FT35

	FT26	FT29	FT31	FT32	FT34	FT35	FT36	Σ
Mean error	0.1775	0.2094	-0.1688	-3.6411	-2.0570	-2.0102	-2.5381	**1.5432**
RMS error	1.7786	1.1419	2.4303	4.4476	4.4030	2.8627	3.7066	**2.9672**

Table 79: Performance evaluation of the altitude control loop over the 4th Flight Test Campaign

Figure 342. Here, the red reference signals are the ones coming from the navigation logic, which were shaped inside the lateral directional control for smooth turning behavior. This explains the visible gaps in the reference value. During flight test (FT) 14 all the loops of the baseline controller were working together, i.e. all the functionalities were tested together. This lead to the successful horserace flight test illustrated in the $N - E$ coordinate frame in Figure 343. Note that this horserace pattern corresponds to

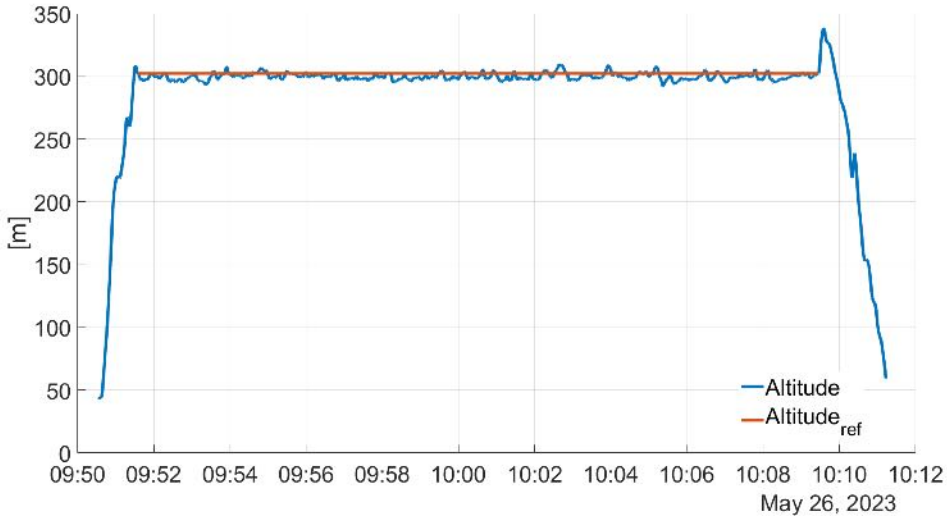


Figure 341: Altitude hold during FT36

the course angle reference in Figure 342.

Figure 342: Course angle tracking during FT14

It can be seen that the baseline controller was able to control the aircraft along the flight test pattern. However, due to the technical difficulties, the concept of the horserace flight pattern was discarded. Instead, circular flights were proposed for the flutter tests, where the airplane keeps a constant bank angle. Accordingly, the lateral directional control loop was not used in the future flight tests.

Autothrottle

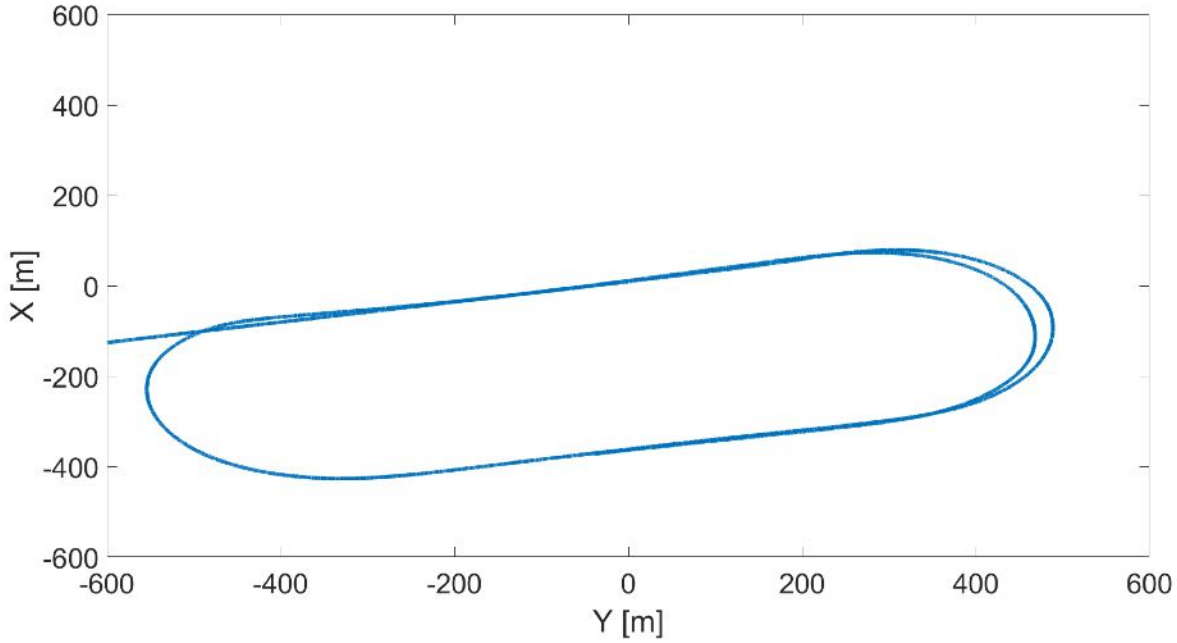


Figure 343: Horserace pattern during FT14

For the autothrottle loop, responsible for controlling the indicated airspeed of the aircraft initially three different control configurations have been developed. This is due to the inexact knowledge of the engine's dynamics and characteristics, which was later refined by the available in-flight measurements. FT12 was the first successful testing of the autothrottle loop, where the three different controllers have been tested subsequently. Two versions of the 2-DOF proportional–integral–derivative (PID) controller have been implemented, where the gains of the 'Robust' version have been decreased, as compared to the 'Performance' version. This was necessary for safety reasons: compromising the tracking performance in order to have a less aggressive control behavior. In addition, a coupled altitude-speed control approach, i.e. Total Energy Control System (TECS) was also designed, implemented and tested. The time-domain flight results can be depicted in Figure 344, while the numerical comparison is given in Table 80.

	Robust	Performance	TECS
Mean error	-1.7520	-0.9852	-0.5419
RMS error	2.0620	1.6283	0.9367

Table 80: Performance of the three autothrottle controllers during FT12

Although the TECS controller was slightly outperforming the 'Performance' and the 'Robust' versions, this architecture has been discarded due to the high amplitude oscillations in the engine revolutions (RPM). This phenomena was also present for the 'Robust' and 'Performance' controllers, but in a less articulated form, however had to be addressed in the future due to its damaging effect on the engine. Nevertheless, the remaining flight tests were only using the 'Robust' and the 'Performance' controllers.

The goal of Flight Test 16 was to further test the authotrottle controller's performance by increasing the airspeed in multiple steps. The collected flight data can be seen on Figure 345. This flight test revealed an important implementation issue: the upper saturation limit of the controller was set incorrectly and

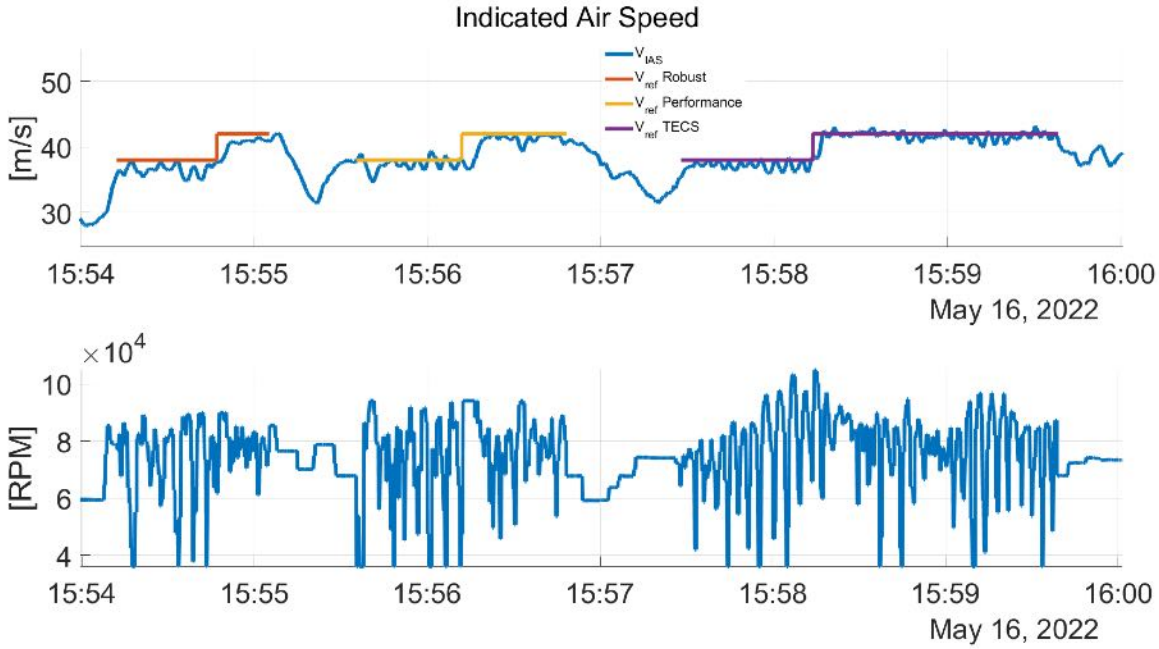


Figure 344: Autothrottle speed tracking and the corresponding control signal during FT12 - Comparison of three controllers

accordingly not all the available control authority was facilitated. This can be clearly seen in Figure 345, where after saturation the aircraft failed to track higher airspeed commands and pilot commands were issued instead.

FT26 was dedicated for the comparison between the three different autothrottle controllers, as in the previous Flight Test Campaigns. Namely, the following configurations were tested:

- Robust: 2 DOF PID controller with reduced gains
- Performance: 2 DOF PID controller with increased gains
- Total Energy Control System (TECS)

The in-flight measurements can be seen in Figure 346, where each controller were commanded the same multistep reference speed command. The numerical evaluation of the three performances showed that the Performance controller provides the most satisfactory and balanced tracking: although the tracking error of the TECS controller was found to be less (0.156) than of the Performance controller (-1.1), but the latter controller provided less oscillation (RMS value 2 in contrast with 3.8). In addition, the input energy and the profile of the commanded RPM signal (RMS value of 6.7×10^4 vs. 7.4×10^4) were more beneficial from the operation point of view of the aircraft. Please observe this phenomena on the lower subfigure in Figure 346. Accordingly, the Flight Test Crew decided to use the 'Performance' controller for the remaining flight tests of the 4th Flight Test Campaign.

Table 81 collects the numerical performance metrics for the selected 'Performance' autothrottle. Considering the seven flight tests, evaluated in the table, a solid and precise tracking behavior can be concluded. We emphasize that from FT29 the flutter configuration were used and later open-loop (FT32-34) and closed-loop (FT35-36) flutter tests were executed. However, neither the numerical values, nor the time-domain measurements contain any significant decrease in the performance of the

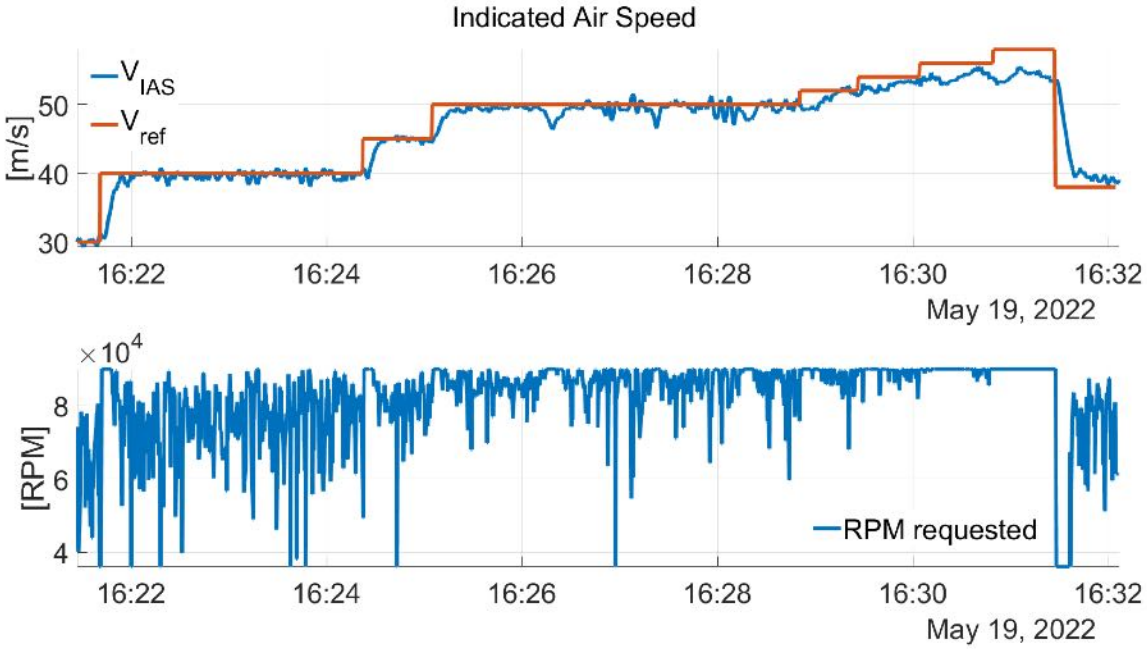


Figure 345: Autothrottle speed tracking and the corresponding control signal during FT16

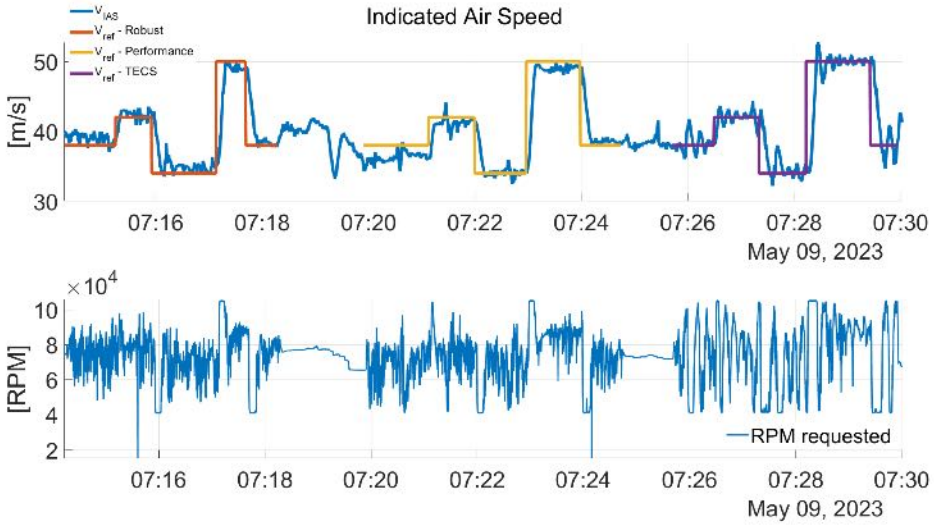


Figure 346: Testing various autothrottle controllers in FT26

autothrottle loop. The $0.61\frac{m}{sec}$ overall absolute mean tracking error is considered to be very precise, even under changing flying conditions. Comparing the recorded signals on Figures 347-349 one can conclude that the autothrottle controller was able to accelerate the aircraft to the commanded reference value, without overshoots. In addition, observation the commanded RPM signals in the lower subfigures of 347-349 shows a decreased oscillation due to the implemented roll-off filter.

Summary

	FT26	FT29	FT31	FT32	FT34	FT35	FT36	Σ
Mean error	-1.1699	-0.0903	-0.1849	0.0256	-0.0868	0.5605	0.2725	**0.611**
RMS error	2.0084	0.8107	1.2227	1.2674	1.9800	2.1104	1.7078	**1.5868**

Table 81: Performance evaluation of the 'Performance' autothrottle loop over the 4th Flight Test Campaign

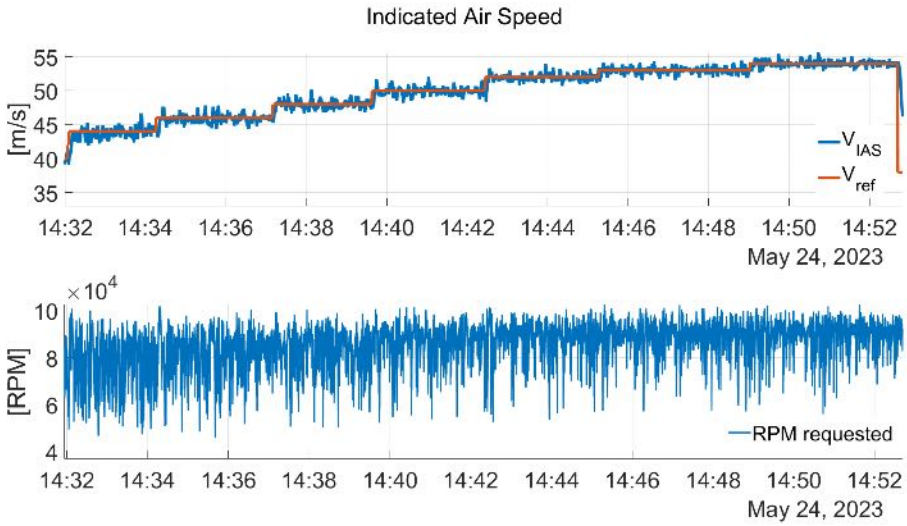


Figure 347: Open-loop flutter speed test during FT32

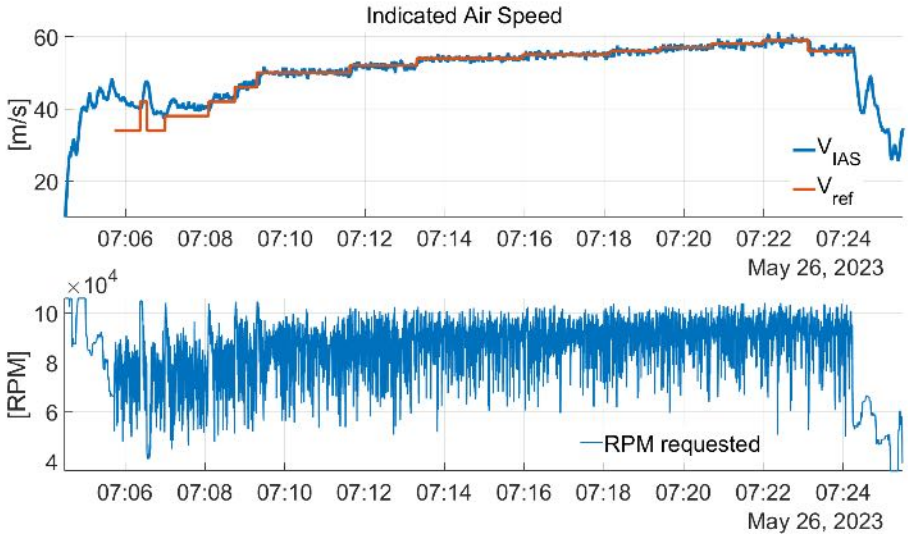


Figure 348: Closed-loop flutter tests with DLR controller during FT35.

In summary, the 1st Flight Test Campaign was sucessful for testing the baseline controller's functionalities, FT14 was the first flight test were the all the different loops of the controller worked together. The most inner loops of the roll and pitch attitude were performing precisely and were also working smooth with the corresponding outer loops of the lateral-directional and altitude controllers. The Flight Test Campaign also revealed some minor issues related to the baseline controller. These were main

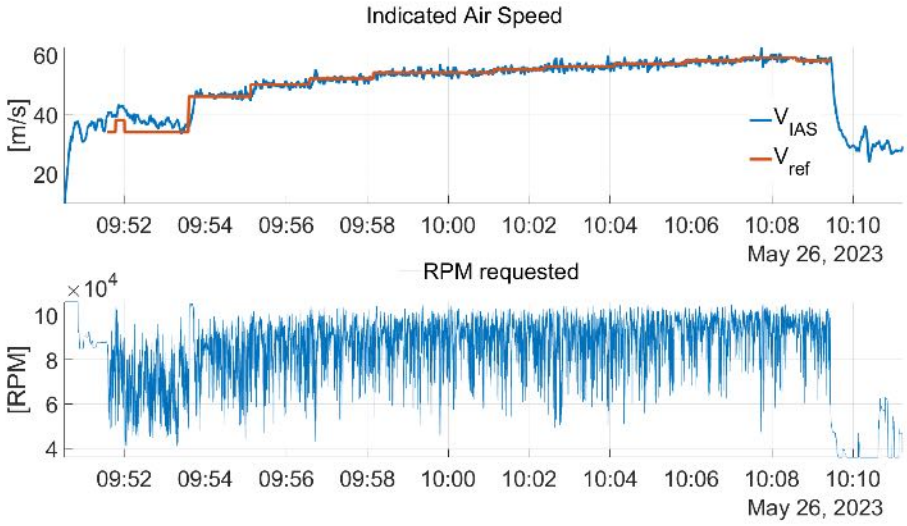


Figure 349: Closed-loop flutter tests with SZTAKI controller during FT36.

related to the implementation of the autothrottle loop: first, the saturation limits had to be fixed, second, the oscillatory nature of the commanded RPM signal had to be eliminated. These issues were fixed in the next flight tests.

Flexible body mode identification (DLR) One important aspect of UAV operation is the identification and characterization of their dynamic behavior, particularly in terms of their flexible body modes. Operational Modal Analysis (OMA) is as a powerful tool for extracting dynamic characteristics of structures and mechanical systems. It involves the use of measured output signals to identify the modal parameters, such as natural frequencies, damping ratios and mode shapes without the need for a dedicated external excitation source. In the context of UAVs, OMA can be applied to determine the flexible body modes that significantly influence the flight dynamics.

The flexible body modes of UAVs are associated with the deformations and vibrations that occur due to the interaction of aerodynamic forces with the UAV structure during flight. These modes are typically lower in frequency i.e. under 20 Hz. Understanding and accurately identifying the flexible body modes is important for several reasons. Firstly, flexible body modes can affect the stability, control, and maneuverability of UAVs. Uncontrolled vibrations induced by these modes can lead to reduced flight performance, increased energy consumption, and even catastrophic failure. By characterizing the flexible body modes, engineers can design control systems that mitigate the adverse effects and enhance the UAVs overall performance. Secondly, the identification of flexible body modes enables the development of structural health monitoring (SHM) systems for UAVs. Continuous monitoring of the modal parameters can provide insights into the structural integrity, detect any damage or fatigue, and aid in proactive maintenance, thereby ensuring the long-term reliability and safety of UAV operations.

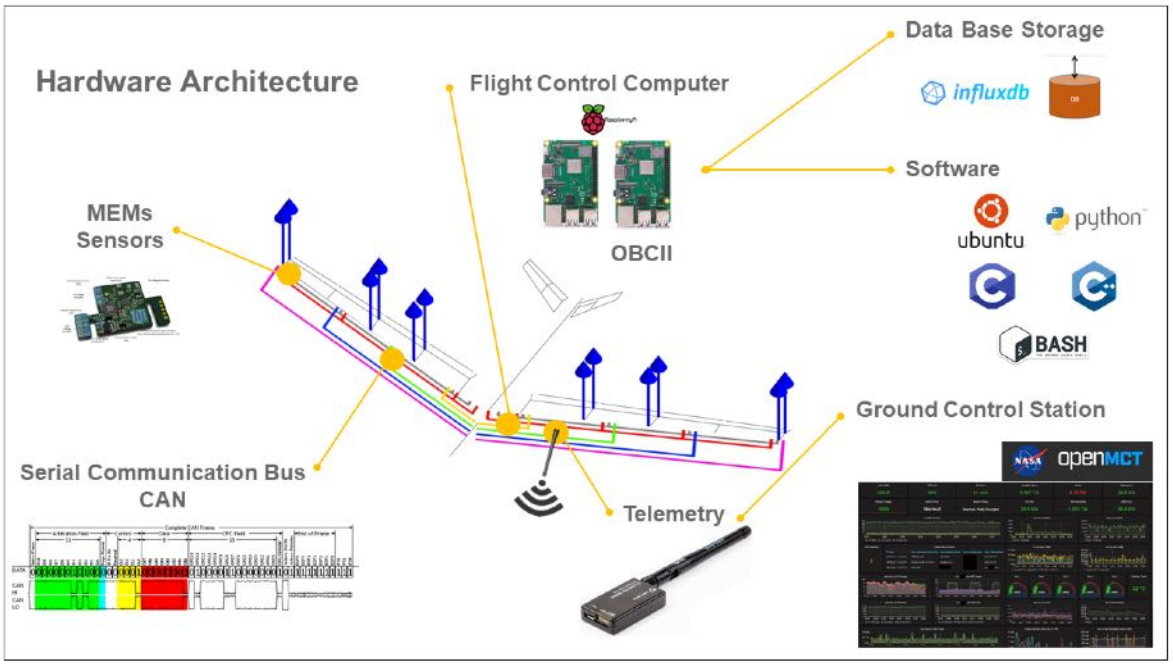


Figure 350: System for online flight vibration monitoring.

To this end a real time capable online monitoring system was developed. The flight testing system for online flutter monitoring is shown in Figure 350. The system architecture was designed to be modular, plug and play with no mandatory user interaction during operation. The system was also designed to be highly configurable for deployment on a large variety of structural layouts of UAVs, from conventional wing-based aircraft to Vertical Take Off and Landing VTOL multicopters. The system is scale-able with different sensor types and communication protocol options to suit different size structures. Furthermore, in order to deploy multiple resource intensive processes such as QR factorisation and Singular Value Decomposition SVD without blocking new data or competing for resources on the miniaturized hard-

ware, the system is setup with multiprocessing. Multiprocessing enables parallelisation by assigning tasks to different CPU cores and threads. In the framework of the FLIPASED project, the system hardware consists of in-house Printed Circuit Board PCB Inertial Measurement Units IMUs which measure acceleration and angular velocity.

Six IMU Printed Circuit Boards PCBs are mounted in each wing at the positions shown in Figure 350. Each IMU PCB consists of a PIC microcontroller, an MPU6000 digital Micro Electro Mechanical Systems MEMs sensor, an AD2286 analog MEMs sensor and a CAN transceiver. Each PCB is mirrored on the top and bottom for full redundancy. The microcontroller requests data from the MEMs sensors at 1 kHz using I2C serial communication, and applies a Finite Impulse Response (FIR) high pass filter. These sensors are connected via the serial communication CAN bus to the FlightHAT. The data is requested on the CAN bus by the host device FlightHAT at a sample rate of 200 Hz. This sampling and filtering strategy has proven effective to cover the frequency range relevant for flutter of this aircraft and to maintain accurate phase information among the sensor signals.

The FlightHAT is a real time embedded controller for collecting data for the Flight Control Computer FCC. A second miniaturized computer was built into the aircraft to enable real time flutter monitoring. A Raspberry pi 4 with quad core cortex-A72 SoC@1.5GHz and 8GB or RAM was integrated into the flight stack with a 3D printed housing. The data is transferred from the FCC to the second Raspberry pi OBCII via ethernet using ZMQ. ZMQ is a lightweight messaging library, which provides standardized messaging over a socket based communication channel. The OBCII receives IMU data in real time, builds a FIFO buffer, performs signal processing, operational modal analysis using Stochastic Subspace Identification SSI and mode tracking using machine learning using state of the art algorithms in Python. Once modal parameters have been identified using SSI, they are tracked using Density Based Spatial Clustering DBSCAN into mode families. Finally, the data is passed to the telemetry system, programmed using Finite State Machine (FSM) logic to encode and transmit the data over the 433 MHz telemetry module. A ground station running the same Python class then receives and decodes the messages. The data is saved to a time series data base called Influx DB and the frequencies and damping ratios are plotted as a function of time in the NASA OpenMCT GUI in order to observe trends of decreasing damping which would indicate the approach to the flutter boundary.

Flight test FT16 was dedicated to aeroelastic system identification. The aim was to identify and track modal parameters in real time at different points of the flight envelope. In order to identify a consistent Linear Time Invariant (LTI) system, constant speed circuits were flown. Furthermore, several observations at the same flight test point reduce the uncertainty in the modal parameters. In order to improve the flight speed consistency, loops were flown with constant bank angles. This allowed the time on test point to increase from 16 - 30 seconds for horse race pattern flight to 120 – 240 seconds.

The acceleration time history of the 12 IMUs during FT16 are shown in Figure 351. The sections are colored according to the flight state as follows:

- Cyan: taxi, takeoff, augmented mode and auto-throttle check.
- Magenta: 30 m/s flight speed.
- Yellow: 40 m/s flight speed.
- Green: 50 m/s flight speed.
- Blue: 55 m/s flight speed.
- Red: airbrakes on, landing and taxi.

Acceleration values of up to 150 m/s2 were experienced on the wings during take-off, landing and taxi. In flight values rarely exceeded 20 m/s2 with rms values of 1.3 m/s2 at 30 m/s, 1.9 m/s2 at 40 m/s, 2.6

m/s2 at 50 m/s and 2.2 m/s2 at 55 m/s. Interestingly the rms values increase almost linearly from flight speeds 30 m/s to 40 m/s to 50 m/s but then decrease at the maximum achieved flight speed of 55 m/s. Turbulent conditions were also experienced during each flight state as seen from the spikes in the time history. In general random response as generated by natural turbulence is ideally suited to satisfy the mathematical assumptions of the unmeasured input forces for the OMA algorithms.

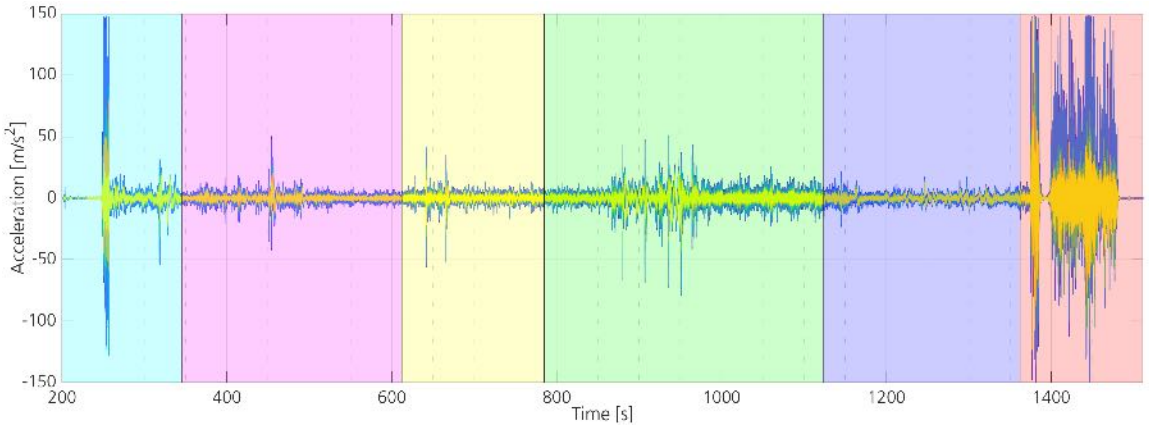


Figure 351: Acceleration time history during flight 16.

Modal analysis was performed in real time on the aircraft on 30 second data buffers with 60 % overlap. The measurement duration of 30 seconds was chosen so as to be short enough to not smear physical changes to the system as well as long enough to contain sufficient observations of the target modes. In theory this data buffer length can be adjusted based on the how well the modes are excited, which is primarily a function of flight speed. However, for this work the buffer length was kept constant. An example of 30 seconds of acceleration data sampled at 200 Hz from the 30 m/s flight speed is shown in Figure 352. Despite a speed variation of less than +- 2 m/s and bank angle variation +- 4 degrees the data is not highly stationary with some gusts (example at 415 seconds) and a slight orientation change in the gravity field (423 seconds). These are challenges which will always be faced with real in-flight data, and should be kept in mind by the engineers when interpreting the results.

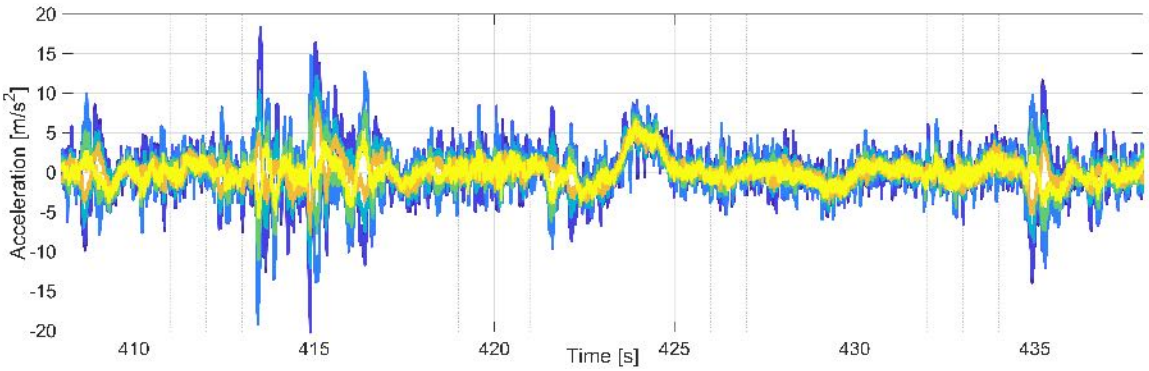


Figure 352: 30 second time buffer.

The cross power spectral densities CPSDs after decimating the data from 200 Hz to 60 Hz are shown in Figure 353. The response spectrum is seen to be relatively flat in the band 0 Hz to 30 Hz, since all modes get additional damping from unsteady aerodynamics, which is more significant than structural damping on ground. This is also an indicator of broadband aerodynamic excitation and supports the use of time domain modal analysis algorithms which are not based on curve fitting in frequency domain.

The stabilization diagram is shown in Figure 354. Despite the relatively flat spectrum, lines of consistent poles at increasing model orders can be observed. Six modes are identified using the Stochastic Subspace Identification SSI algorithm from the current data buffer between 0 Hz and 30 Hz. The block size was set to 16 and the maximum model order to 80. The modes which match within a given frequency, damping and eigenvector tolerance belong to a unique mode family and are given a unique color. This mode color will be used consistently throughout this section – i.e. mode 1 will always be red. Finally, the measured average spectra is plotted by the blue curve and the synthesized spectra by the red curve. This shows how well the identified model recreates the measured data in the frequency domain. The fit is seen to be in agreement, but is not perfect. The use of additional sensors on the tail and fuselage as well as measurements in the x direction on the wings are expected to improve the modal model and therefore the synthesis error.

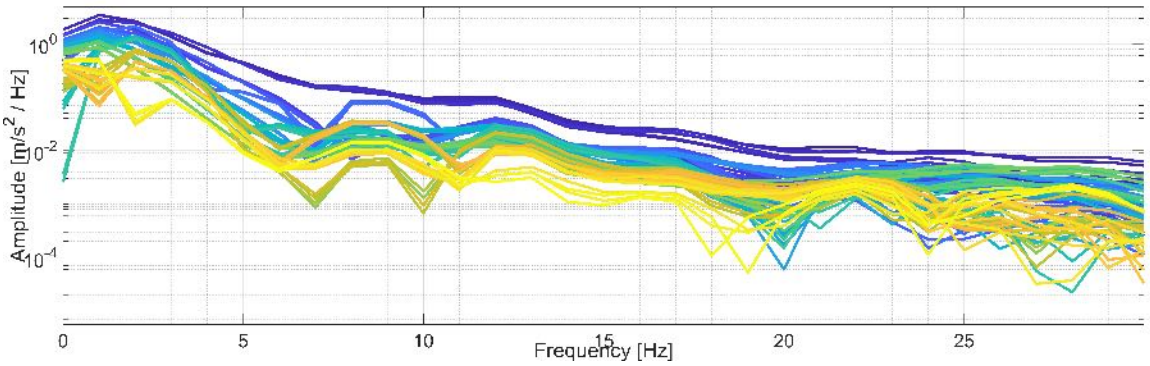


Figure 353: Cross power spectral densities.

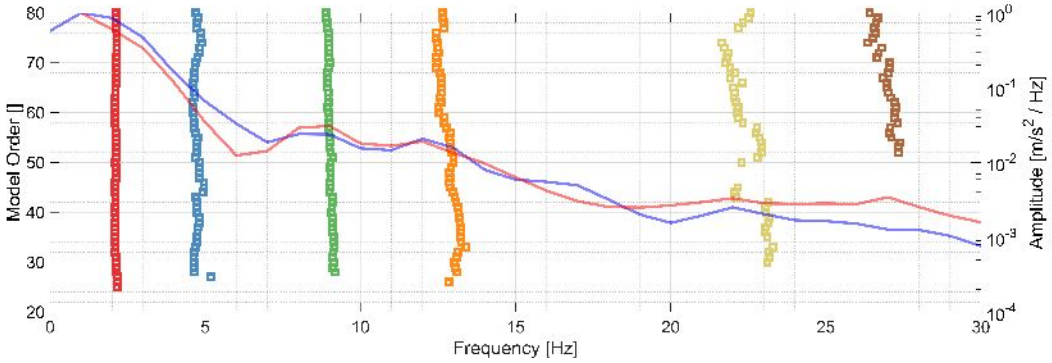


Figure 354: Stabilization diagram from SSI.

The modal assurance criterion MAC shown in Figure 355 is the normalized dot product of the eigenvectors and shows that the six identified modes are linear independent. The eigenvectors or mode shapes of the six modes are shown in Figure 356. The first mode at 2.1 Hz is the rigid body roll mode. The second mode at 4.7 Hz is the 2n wing bending. Next come 3n wing bending at 8.9 Hz, 4n wing bending at 12.6 Hz, 5n wing bending at 22.6 Hz and 6n wing bending at 26.4 Hz. Wing torsion was not identified since it was designed to be above 60 Hz for this baseline wing set. The phase purity of the mode shapes is high and the shapes are clean and symmetric. This is an impressive result considering that the data is measured with MEMS sensors, processed with light weight low cost embedded systems, on short data buffers, in real time during actual flight conditions. Unfortunately due to the lack of sensors on the fuselage or V-tail only wing bending modes could be identified.

The aim of this system is not simply the real time identification of modal parameters, but also the

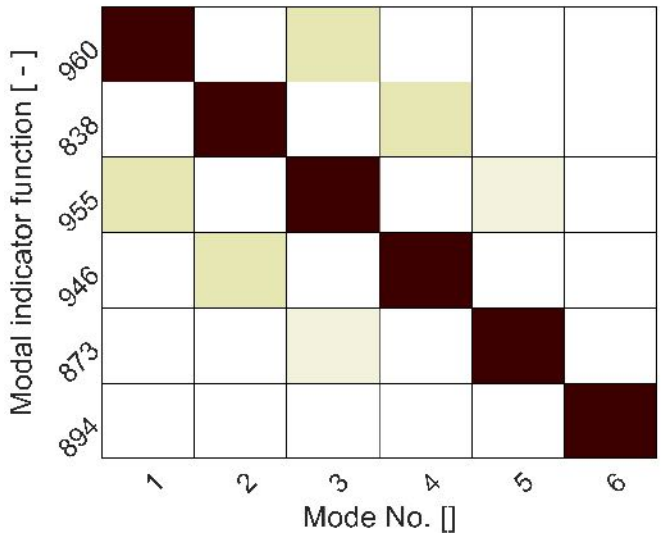
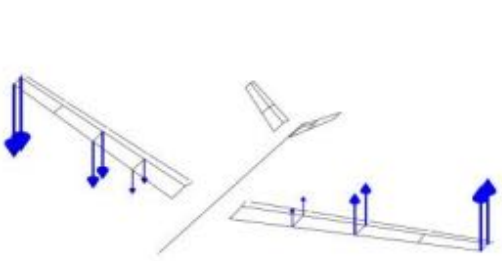


Figure 355: Modal assurance criterion MAC.

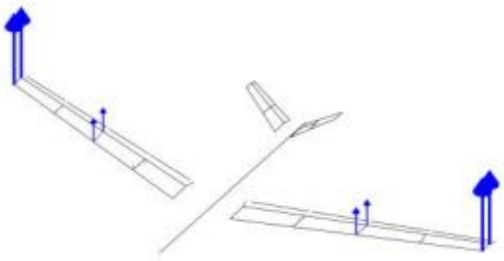
tracking and quantification of these parameters through the flight envelope. The resulting frequencies and damping ratios of the six mode families are plotted throughout the flight in Figure 357 and Figure 358. The six mode families are represented by uniquely coloured squares as a function of mode set. Each mode set consists of a 30 second data buffer with a 60 % overlap. The flight speed is indicated by the blue line and the flight altitude by the dashed black line. Here the constant speed circuits of 30 m/s, 40 m/s, 50 m/S and 55 m/s can be seen. The flight altitude remained fairly constant at 400 m. In Figure 357 six horizontal lines of uniquely coloured squares representing mode families can be observed. It can be seen that the frequencies vary during the flight and that not all modes are identified in all mode sets. This is expected to be because of the different excitation conditions resulting from flight speeds and wind conditions. The damping ratios show larger variations especially for mode 1 and 2. The identified aeroelastic damping in the range of 10 - 20 for modes 3 - 6 are in the range of expectations from engineering judgement. The question which then arises is whether trends can be seen in the modal parameters as a function of the dominant physical factors.

The frequencies and damping ratio's are therefore plotted as a function of flight speed in Figure 359 and 360. Here the clusters of modal parameters at each flight speed are averaged. Firstly it should be noted that the test was conducted on the baseline wingset (-0 wing). This wingset was designed to be completely stable in the flight envelope, and we therefore do not expect to see flutter. Nevertheles some changes in the modal parameters are expected. The eigenfrequencies show a general increase from 30 m/s to 50 m/s. The damping ratios of modes 2 – 6 also increase from 30 m/S to 55 m/s. The rigid body roll mode shows a reduction in damping ratio from approx. 60 % at 30 m/s to 30 % at 55 m/s.

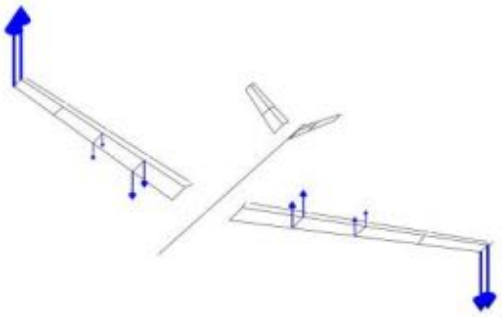
In conclusion, flight testing of the UAV demonstrator T-FLEX in the EU project FLIPASED successfully demonstrated the capabilities of an online modal identification system for flutter monitoring based on miniaturized hardware. The system was deployed in an extensive flight test campaign at the DLR Cochstedt airport. The flight test campaign provided valuable data and thoroughly tested the capabilities of the system. The online monitoring system integration into the flight control system of the aircraft performed well. The on board signal processing, modal analysis and mode tracking produced accurate results and ran in real time. Furthermore, the in-house developed telemetry system proved stable and robust with zero fatal disconnects and no package losses. The system was able to identify and track modal parameters in flight, and flight test engineers were able to visualise the frequencies and damping



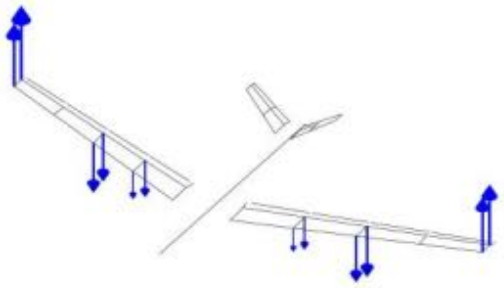
(a) Mode 1 - rigid body roll



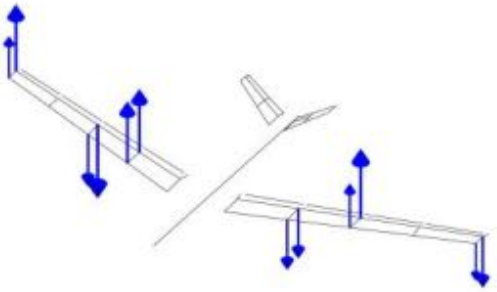
(b) Mode 2 - 2n wing bending



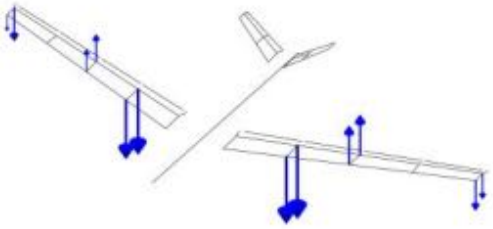
(c) Mode 3 - 3n wing bending



(d) Mode 4 - 4n wing bending



(e) Mode 5 - 5n wing bending



(f) Mode 6 - 6n wing bending

Figure 356: T-FLEX mode shapes.

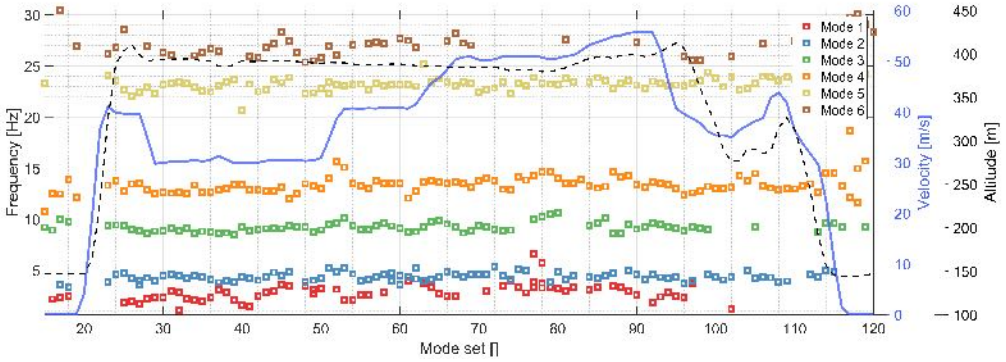


Figure 357: Mode tracking of eigenfrequency during flight 16.

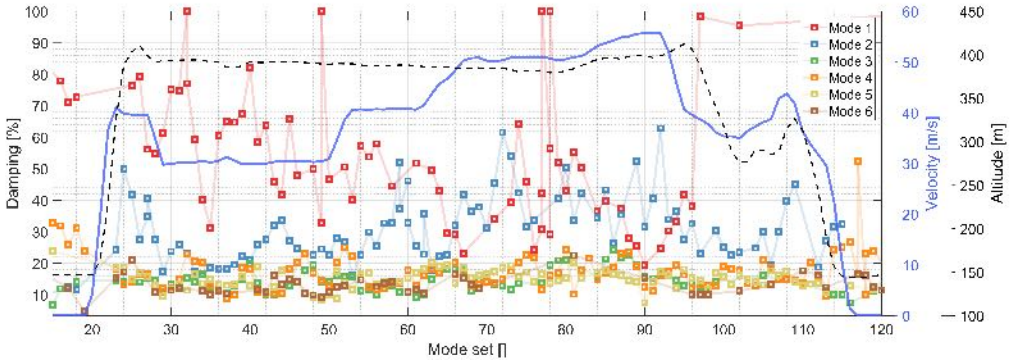


Figure 358: Mode tracking of damping during flight 16.

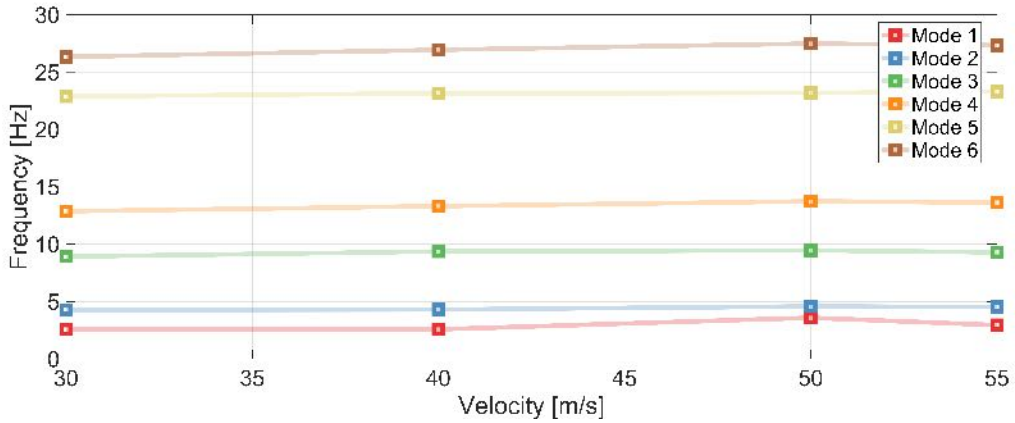


Figure 359: Mode tracking of eigenfrequency as a function of flight velocity.

ratios as a function of time or velocity at the ground control station. Six wing bending modes were identified and tracked during the flight campaign. The six modes showed trends of increasing frequencies and damping ratios for all the elastic modes. This was in agreement with the non-linear aeroservoelastic model. The system has therefore been demonstrated as a capable and reliable tool for real time flutter monitoring during flight testing. Improvements to the system regarding additional sensors in the

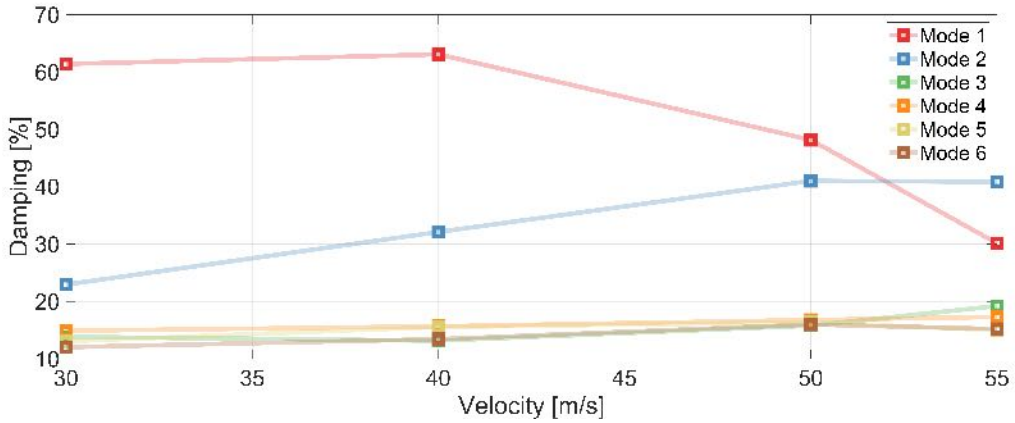


Figure 360: Mode tracking of damping as a function of flight velocity.

V-Tail as well as in-plane measurements in the wings are underway. Furthermore, testing the online monitoring system on the flutter critical wing set will provide the ultimate test of the result accuracy in predicting the flight envelope. Finally, the integration of the online modal analysis system with the onboard flight control system for active flutter control will be the next step in the current research and development.

Aeroelastic flutter is a phenomenon that occurs when the interaction between aerodynamic forces and structural dynamics leads to self-excited vibrations of an aircraft. It is a critical phenomenon that can have severe consequences if not properly understood and mitigated. Flexible body mode identification using Operational Modal Analysis (OMA) plays an important role in understanding and identifying aeroelastic flutter.

In the context of the flutter critical wing in the FLIPASED project the identification and tracking of the flutter critical modes is important for the following reasons:

1. identify the flutter critical modes and track them through the flight envelope
2. confirm whether simulation results of a symmetric flutter coupling is correct
3. provide data driven results for comparison and updating of the aeroelastic simulation model
4. quantify the effect on the damping of the active flutter controllers
5. ultimately verfiy the design goal of a flutter mechanism

To this end, the identification and tracking of the flexible body modes using OMA in real time allowed engineers to assess the stability of the UAV as detailed in FLIPASED Deliverable D3.6. Since the previous tests were conducted on an aeroelastically stable baseline wing, the results of the DLR OLM tool on the flutter critical wing would provide the ultimate test of the algorithm performance. The results of the OLM tool are presented in Section 3.6 for the open loop sub-critial case, 3.7 comparing the active flutter controllers in the sub-critical case and finally in 3.9 with open loop tests to the flutter critical speed.

Baseline aerodynamic analysis (TUM) Take-off performance

After analysing all the take-offs performed during the 1st test campaign (Figure 361) it was discovered that even though the elevator is down all the time during the take-off, $2-3s$ before actual take-off point the airplane lifts the tail anyway (note the pitch angle). It is suspected, that this is the point where the ground control is lost. Such a conclusion would also correlate well with the data collected during taxi tests, where it was noted that during the high-speed phase the aircraft becomes very uncontrollable.

This indicates that the tail authority is not enough. As the angle of attack cannot be kept at $4deg$, it could be the reason why the aircraft always lifts-off late with around $30m/s$. Similar pattern can be seen in the previous flight tests as well.

It is suggested to either increase the elevator authority, or use outboard flaps to get more pitch-up moment for the upcoming flights. In such case it could be expected to decrease the TO distance by around $2-3s$ (around 50-70m), and reduce the airspeed by around 6-8m/s.

Lift and drag polars (TUM)

Flight tests 10, 16 and 17 were used as a basis for a publication ”In-flight drag measurement and validation for a medium-sized UAV”[5]. The article presents how the flight test data was used in order to model the lift and drag characteristics of the baseline demonstrator.

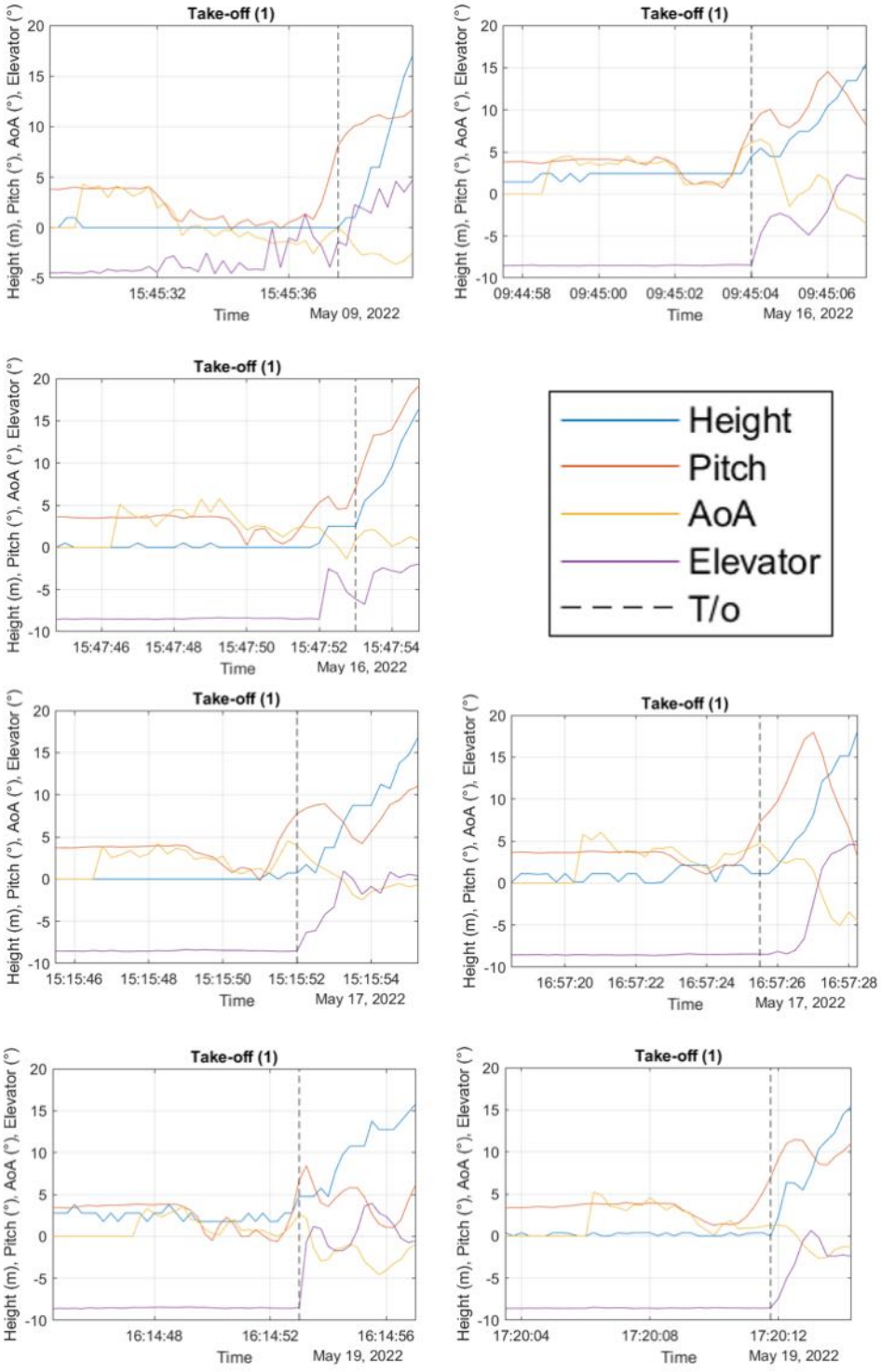


Figure 361: Take-off performance for the 7 test flights. Note the ”pitch bucket” visible for 2-3s before the actual take-off point (marked with the dashed line) and the elevator command (in purple).

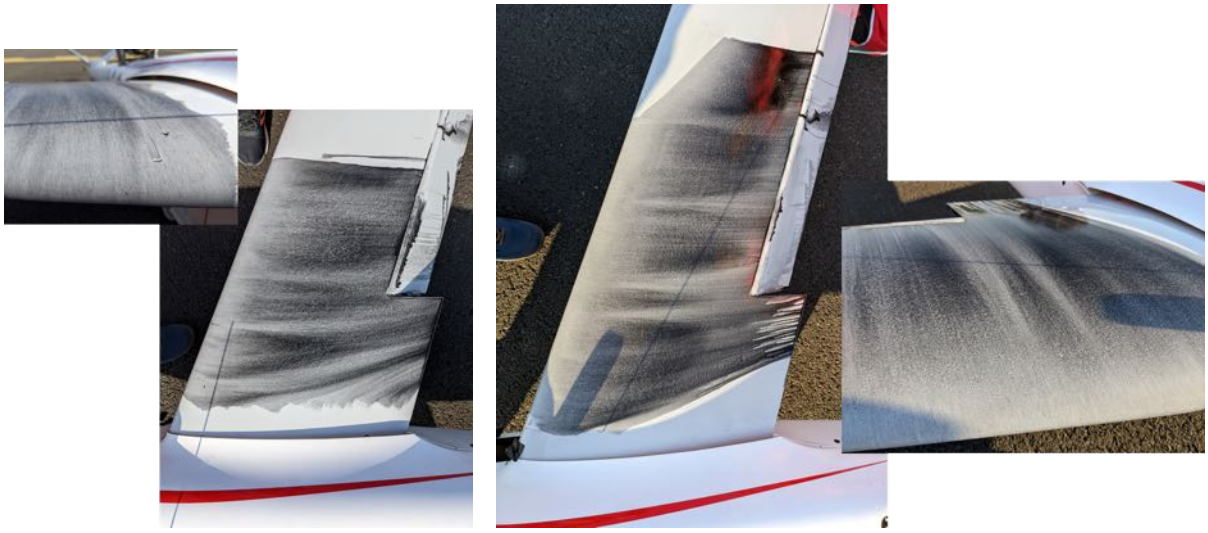
Airflow visualization over the wings (TUM) Airflow visualization experiments with oil flow were performed to investigate the boundary layer effects in-flight.

A 1:1 mixture of Mobil 1 ESP 5W-30 engine oil and Liqui Moly SAE 75W-90 high performance gear oil was mixed with black pigment [15]. The resulting viscosity of the mixture seemed to fit the airspeeds encountered in flight.

Two flights were done with the mixture applied to the root area of the upper wing - flight tests 13 and 14 (Figure 362a). It was envisioned to be able to record the transition of the oil in-flight, but the resolution of the camera was too low. It was noticed, however, that the pattern stabilizes after around 10min of flight.

After the first test on FT13 there were clear identifications of boundary layer transition due to particles on the surface after. The cross-flow component inside the boundary layer was also visible towards the trailing edge of the wing.

A bigger area was covered with oil for the second test, and a decreasing chord length was covered on the outboard section to investigate the effects of the oil itself (Figure 362b). Again, clear transition lines were visible after flight. Though it was not completely clear where do all the turbulent wedges come from. It was postulated that maybe it is due to the wing ribs distorting the wing surface (Figure 363, but the hypothesis could not be confirmed.



(a) Flight test 13. **(b)** Flight test 14.

Figure 362: Oil flow patterns after two test flights.

There was a bigger conglomeration of oil visible on the outboard part of the covered area where the decrease in covered chord length started. Even though a similar pattern would be expected during a laminar separation bubble, it could not be confirmed that that area simply had more oil than the root area.

Even though the experiment seemed to provide visible results, there was not enough time available to prepare the oil for every flight. Therefore, no definite conclusions about the boundary layer effects can be made.

The following lessons can be noted for the future:

- The viscosity of the mixture seemed perfect for the Reynolds number regime of the T-FLEX.

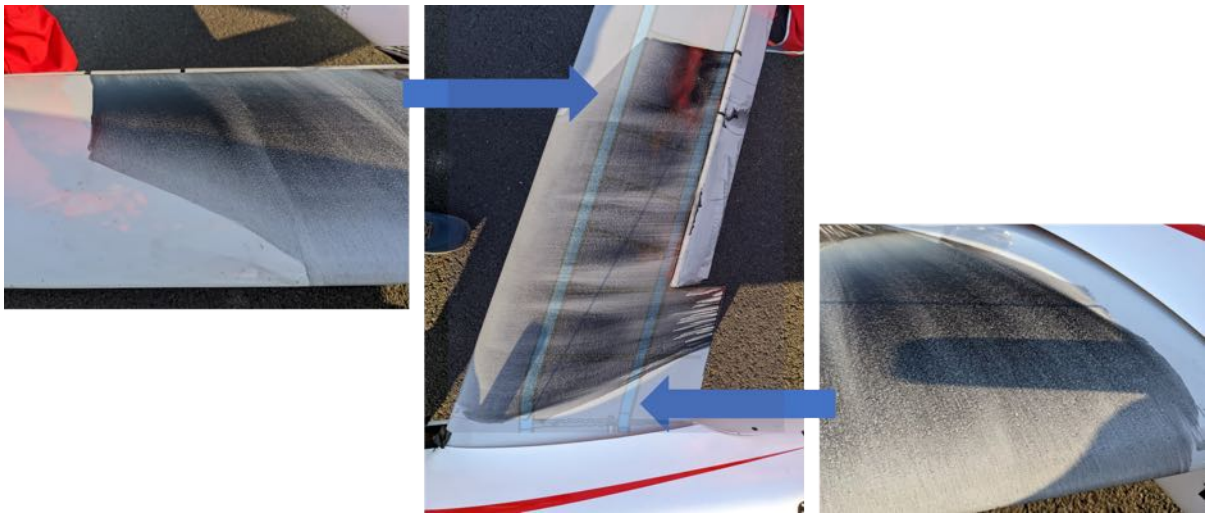


Figure 363: Oil flow pattern after flight test 14 with wing ribs overlayed in the middle.

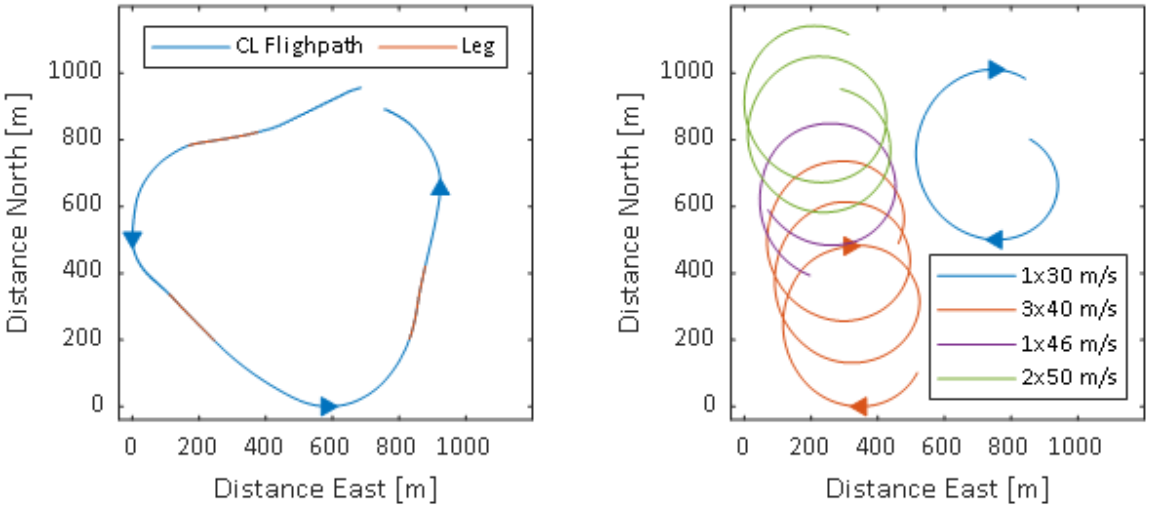
- Pattern stabilises after 10min of flight.
- Colour was wrong. Black was too dark during a sunny day which produced too much contrast. The details were not visible on the camera in-flight.
- 360 camera resolution is not good enough to capture the details, only the overall picture at specific sun angles.
- A turbulator should have been glued in the measurement area for comparison.

Airspeed calibration in-flight (TUM) Collecting correct data about the aircraft's performance is crucial. However, measurements often contain errors that must be accounted for. The focus of this section is to calibrate the airspeed and altitude measurements of the T-FLEX. By analyzing flight test data, wind speed and direction are determined, as well as position errors caused by the pitot static system. This corrected data can then be used for further development and other performance reviews. Three main airspeed calibration methods are tested and assessed during simulator flights and a flight test campaign. Subsequently these methods are implemented into a flight test data analysis tool in MATLAB. Additionally, the position error is compared to an existing Computational Fluid Dynamics (CFD) simulation to match the results of simulations and test flights.

The airspeed calibration methods that were found applicable for T-FLEX are the cloverleaf method, the level turn and the turning acceleration. The calculations used in the level turn method and the cloverleaf technique are similar and simple regression techniques are applied for both. The trajectories flown for these methods are easy to follow (Figure 364). Subsequently, an external software is used for analysis of the data gathered for accelerated turn (Figure 365).

The resulting airspeed corrections are presented in Figure 366. It is visible that all three methods predict a small, from -0.5 to -2.5m/s correction that should be applied to the indicated airspeed. This means that the air-data system indicates airspeed that is higher than the real one. However, taking into account that the accuracy, claimed by the manufacturer, is either 1 percent or $1m/s$, the errors can be considered small.

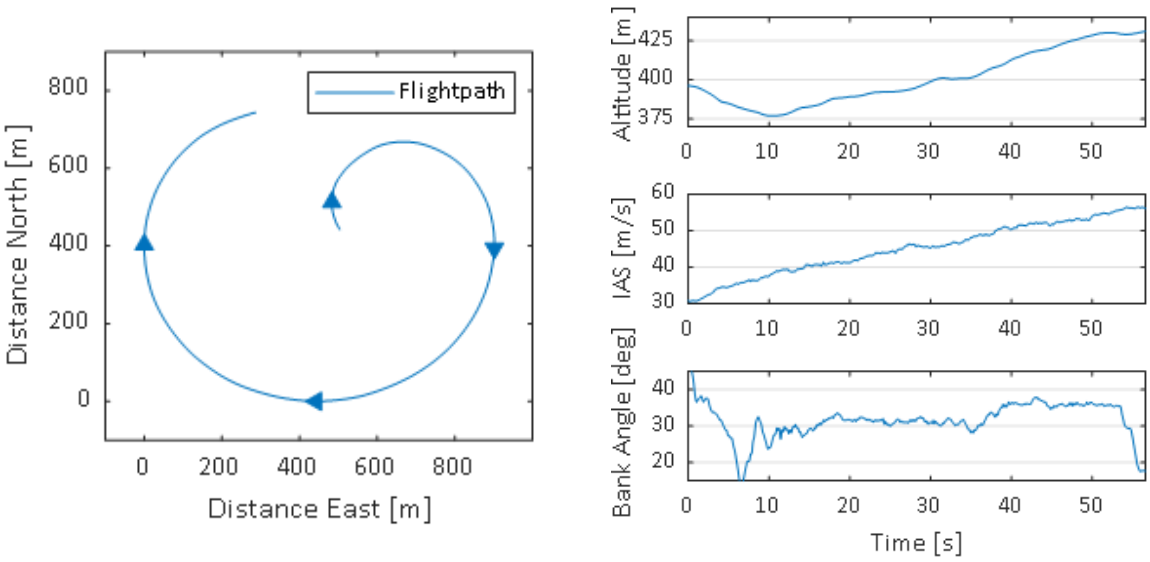
For more details about the investigations done in airspeed measurement error, please consult [93].



(a) Cloverleaf during flight test 11.

(b) Circles during flight test 16.

Figure 364: Examples of the trajectories used for the cloverleaf and the level turn airspeed calibration techniques.



(a) Accelerated turn during flight test 13.

(b) Altitude, indicated airspeed and bank angle during the accelerated turn.

Figure 365: Data used for the accelerated turn method.

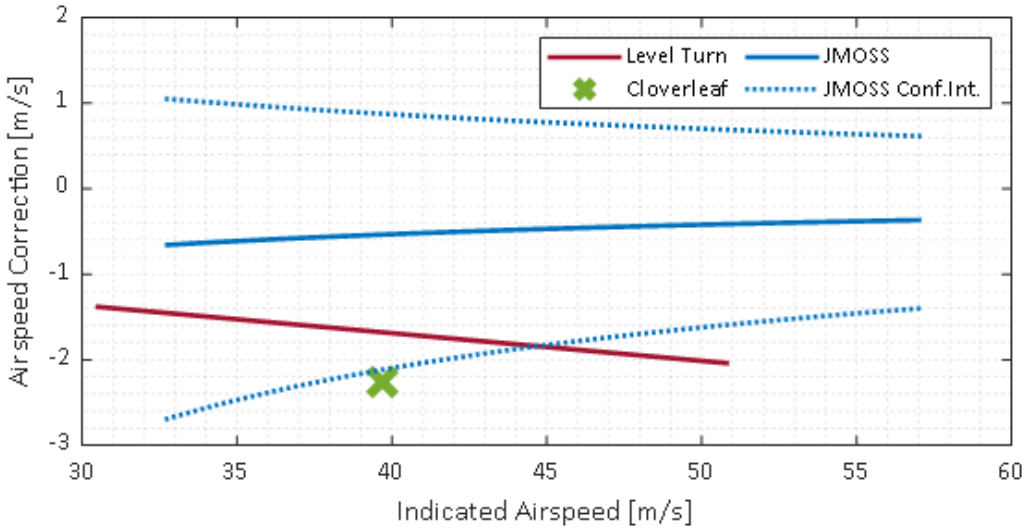


Figure 366: Comparison of airspeed corrections obtained from the level turn method, the cloverleaf method and the accelerated turn method. Note that here JMOSS means the accelerated turn method.

Induced drag experiments (TUM, DLR) Purpose of the induced drag experiments is to validate the drag measurement system and the simulation tool for the wing shape control function. The aerodynamic method employed to validate the induced drag is an enhanced version of a Vortex Lattice Method (VLM) [35] as described in detail in [48]. This extended VLM is implemented in the simulation environment VarLoads [36, 47], which is capable of trimming a free flying flexible aircraft using a mean axes based coordinate frame.

The integrated modelling approach of the simulation tool VarLoads accounts for effects on the induced drag due to flexible deflections as well as due to trim settings. This makes a tedious coupling of the aerodynamic code with the structural solver and a trimming routine superfluous. The objective function for minimizing, respectively maximizing the drag is the thrust setting of the engine of the integral model.

The main objective of the flight test was to compare a high drag configuration to a nominal configuration with zero flap deflection. The simulations showed good agreement with intuition, as a crocodile tooth style configuration showed an major increase of the drag coefficient in the simulations. One constraint from a piloting perspective was not to significantly change the pitching moment of the aircraft. Therefore, the deflections of the outermost flaps were adapted to minimize the change in elevator settings, reflecting a minimal change in pitching moment, while substantially increasing the drag values.

The flap setting selected for the high drag configuration was $+10/-10/+10/+5$ degrees (Figure 367).

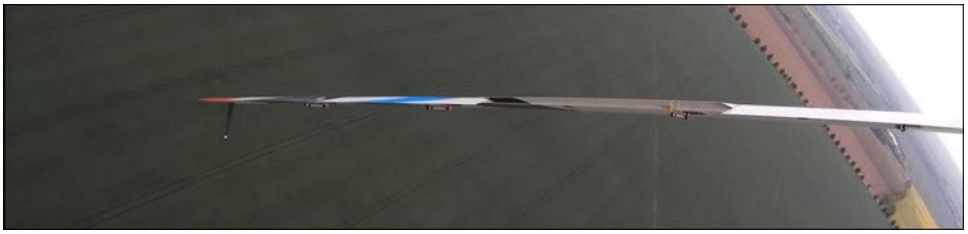


Figure 367: Drag flap configuration during a test point. The flaps here are deflected $+10/-10/+10/+5$ degrees.

Investigation of the accident during the FT23 (TUM) On 29th of August an uncommanded parachute release was experienced with the demonstrator. The aircraft landed outside the airport zone and a fire broke out. The bigger part of the aircraft has completely burnt down. Luckily, all the data sources were available - the main flight log, the two transmitter logs, a log from the secondary on-board computer, the videos from the two tail cameras and the video from the 360 camera, which has fortunately fell out of the fuselage right at the moment of impact.

It could be immediately confirmed from the main flight log that the parachute was released as a programmed safety feature when the reception to the pilots is lost for at least 1.5 seconds. During the flight, the reception was lost for 2s. Furthermore, it became clear that the engine did shut down when the parachute was released, just like planned. But it was not clear why did the engine start again when the aircraft hit the ground. Therefore, the following main questions were raised for the investigation:

- Why was the connection to the aircraft lost?
- Why did the engine start again?

The first question is answered in Section 2.3.3.

At the moment of reception loss, at least some of the transmitter antennas had direct line of sight to the transmitter (Figure 368). The transmitter reception quality is logged by the transmitter itself. It is

communicated to the pilots via audible warnings. However, the logs of the reception quality was never investigated in connection to the position or orientation of the aircraft. This investigation was done by the Flight Test Manager and communicated to the rest of the Consortium in an email, provided in Section 2.3.3.

Figure 368: The moment of reception loss. Here the antennas are visible at the end of the V-Tail and the pilots stand exactly on the point of the runway where the V-Tail is pointing to.

Timeline of the accident

Below is the timeline of the accident. The start of the timeline is the moment when the engine start command is sent before the flight.

- 00:00 - Engine start command is sent
- 09:54 - Last turn initiated
- 09:56 - Jeti loses reception, aircraft switches to Graupner; aircraft switches to failsafe (landing configuration visible)
- 09:57 - Tail cone flies off, drogue chute is released
- 09:58 - Jeti back in control
- 09:59 - Engine is turned off due to parachute release
- 10:00 - Parachute is completely out
- 10:32 - Aircraft crashes on the ground
- 10:42 - Engine start command is sent again
- 11:10 - Engine starts
- 12:02 - Fire visible behind the airbrakes
- 12:30 - Engine turns off
- 14:32 - Smoke visible from the payload bay

- 15:44 - Fire visible in the payload bay
- 16:08 - Complete fuselage is on fire
- 24:24 - The TUM team arrives at the landing site

Figure 369 display the timeline of the accident as captured by the 360 degree camera.

Figures 370 displays the crash site and the remains of the aircraft.

It took roughly 14 minutes for the TUM team to get to the crash site after the aircraft has hit the ground. This was due to the fact that the crash site was outside of the airport area and the team had to drive around the airport on unpaved roads. It is visible from the timeline, however, that already 5:30min after the crash the complete aircraft is on fire. Therefore, the chance to save any bigger part of the aircraft after a fire broke out would have been very unlikely.

Investigation about the engine restart

As mentioned before, it was not immediately clear why did the engine restart after the crash. It was thought that when the parachute gets released, it also triggers a fuel pump shut-off valve, which prohibits any further fuel being pumped into the engine. It was recognized, however, that the valve is not shut permanently if the trigger command is very short. In this case, the command was lost for mere 1.5s, after which the chute release command was commanded. This triggered the fuel valve as well and turned off the engine. But as the command was regained 0.5s later, the fuel valve was reopened.

Another mistake was found with respect to the fail-safe setting on the secondary transmitter. Due to the architecture of the control system, for the Jeti transmitter to be able to switch to Graupner, the engine command has to be set to positive. This was required due to the lack of separate channels on the transmitter communication line. In this case it was envisioned that if the main transmitter loses reception, it switches to the secondary transmitter. Due to a human error, the same fail-safe setting was programmed on the secondary transmitter. This still meant that in a parachute-release case the engine would turn off due to the fuel valve being shut. This is what happened in the current case.

However, as all the risk analysis trees ended with the moment when the parachute was released, it was not envisioned that the reception could be regained after the fail-safe was initiated. In the current case, the aircraft regained reception with both transmitters while going down with the parachute. In combination with the fuel valve being open (due to the very short period of the fail-safe) this meant that the engine could be restarted. But at this point the pilots did not issue an engine start command. It was the second loss of reception when the aircraft hit the ground and the secondary transmitter went into fail-safe that issued the engine start command ten seconds after the impact.

It is speculated that if not for the unfortunate coincidence described above, the fire would not have started after the aircraft hit the ground.

Email about the T-FLEX Reception investigation, sent on 9th of September by Julius Bartasevicius (edited)

I investigated the reception quality of the Jeti transmitter by looking at the GPS data, at separate and combined receiver (Jeti has two) qualities, at aircraft bank angles, distance from the pilot and altitude. I used data from the last two test campaigns for that (excluding two flights because they had many GPS outages). Note that the antennas are pointing to directions 55deg (RX2, tail) and 150 and 210deg (RX1, fuselage).

I would summarise the findings below:

1. No connection to reception quality and a specific GPS position can be found. Therefore I would

(a) 09:57 - The drogue chute release.

(b) 10:00 - Full parachute release.

(c) 10:32 - Front part of the fuselage just after impacting the ground.

(d) 10:34 - Fuselage laying down, the main parachute is visible coming down in the background.

(e) 12:02 - Fire breaks out in the rear part of the fuselage.

(f) 15:25 - Fire breaks out in the front part of the fuselage.

(g) 18:24 - The complete fuselage is on fire.

(h) 21:16 - Parts of the fuselage are already burnt down.

Figure 369: The timeline of the crash as recorded by the 360 degree camera.

(a) The crash site right after the fire was extinguished.

(b) The remains of the payload bay. The batteries have already been removed.

(c) The middle section of the fuselage. Burnt engine and landing gear visible.

(d) The tail section of the aircraft.

(e) The middle section of the aircraft.

(f) The crash site after removing the debris.

Figure 370: The crash site after flight test 23.

not say that there is interference from some antennas around.

2. Reception highly depends on the elevation from the pilot. Elevation angle is the vertical angle from the pilot to the aircraft. When the elevation goes under 13deg, problems can be expected (signal quality of less than 30 percent, Figure 371). Elevation of 13deg means 230m altitude at 1000m distance or 350m altitude at 1500m.

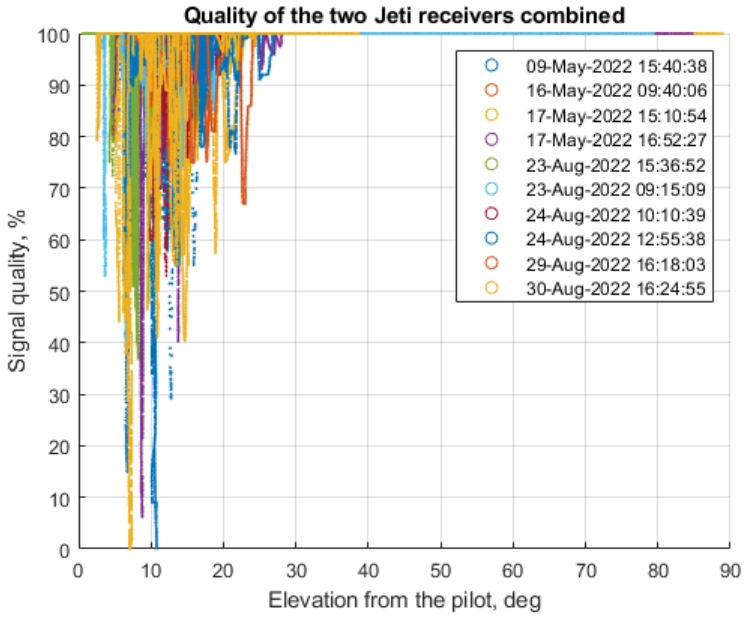


Figure 371: Quality of the two Jeti receivers combined. Note that there is another day when the reception quality went down to zero. At that moment either the reception with the secondary transmitter was still good or the period of no-reception was less than 1.5 seconds.

3. It also depends on the lateral angle to the pilot in the aircraft-fixed reference frame. Below is the polar plot showing the quality for each receiver (Figure 372, aircraft nose pointing to the screen; 0 means pilot is above the aircraft, 90 is when pilot is facing directly the right wing). Here I took only flight moments when the aircraft was perpendicular to the pilot (in a top down view the wings were pointing to the pilot, plus minus 20deg) to reduce it to a 2D problem.

   So there is this area of 45-100deg and 270-320deg where reception is worse. If the aircraft is roughly above the pilot (120-240deg), then reception is good. There is no data in the range 330-30 (we don't fly inverted).

4. If the two above are combined, it's visible that the problems with reception are only in right hand turns (Figure 373, angles to the pilot more than 0).

5. I also tried to see the if bearing from the aircraft to the pilot shows some trends (while keeping the bank angle relative to the pilot within some range, Figure 374). The range from 300 to 0deg does seem more problematic for RX2 (meaning the pilot is in front of the aircraft). This might make sense, since the RX2 is in the tail of the aircraft.

Finally, applying this to the terminated flight: At the moment of the loss of reception we flew at elevation angle too low while also doing a right turn, which put us in the very unfavourable reception envelope.

Conclusion of the investigation

Following the email sent on the 9th of September, the investigation of the crash was concluded. In summary, it could be said that:

- For this specific aircraft there was a small reception blind spot when a right turn is done.
- Reception quality was also lower because of the low flight altitude (below 13deg of elevation). This resulted in a short reception loss with the aircraft.

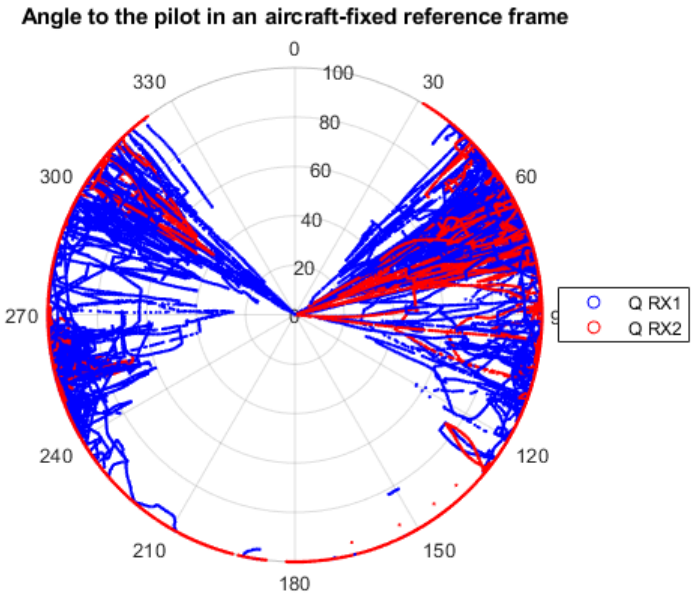


Figure 372: Polar plot of reception quality in relation to the aircraft. Aircraft nose pointing towards the screen.

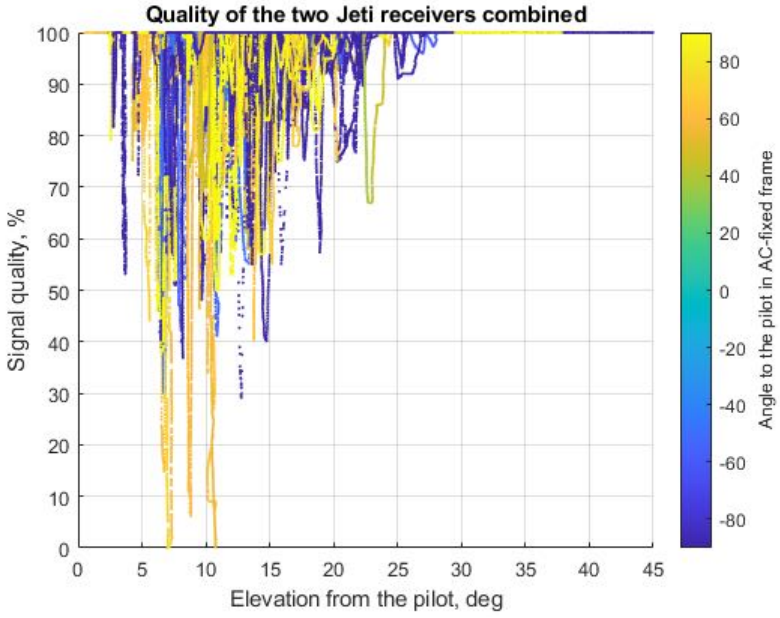


Figure 373: Quality of the two Jeti receivers combined.

- Risk analysis tree was not investigated further than the moment of parachute release.
- Wrongly programmed fail-safe and the short loss of reception meant that the engine could be restarted after the aircraft lost reception for the second time, when already on the ground.
- The restart of the engine initiated the fire, destroying the bigger part of the fuselage.

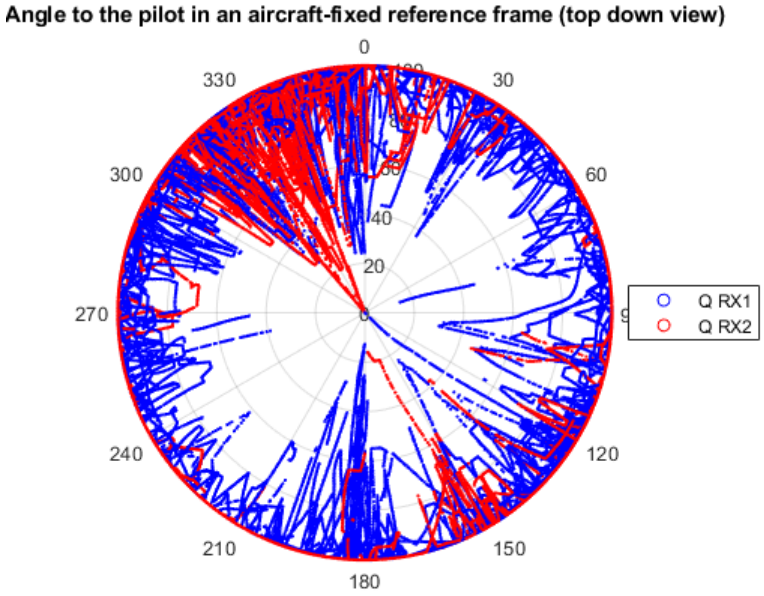


Figure 374: Polar plot of reception quality in relation to the aircraft (top-down view).

Model updating using multiple flight test data (DLR, TUM) Model updating refers to a series of tasks required to estimate or update parameter values of a fixed model structure, usually containing a set of differential equations, that is likely to explain the physical behaviour of a particular system. Consequently, model updating is often preceded by system identification or handled within system identification. Our primary focus is on updating methods of the aeroservoelastic (ASE) models by incorporating data from both ground and flight tests to better predict the aircraft aeroelastic response and thus the onset of the flutter. As part of the FLiPASED project, the application of a new updating method is demonstrated by means of both the discrete state-space model and access to flight test data of the FLiPASED aircraft [128], [129].

The formulation of the proposed updating approach enables correction of the parametric system matrices A, B, C, and D of the initial linearized discrete-time (DT) state-space system derived from a flexible aircraft model. The updating method addresses linear estimation problems. Thus, two formulations for the error minimization of output residuals between flight test data and model predictions have been defined. The first utilises the state-space system's output equations, whereas the second requires both the state and output equations. The methodology for updating that will be described here is based on a linear least-squares approximation resp. a minimum norm solution. The algorithm consists of three steps. In the first step of the algorithm, the calculated states from the initial model corresponding to the rigid body aircraft dynamics will be adjusted using output equations. Here. the considered states are measured and comprise a subset of the output. In the second step, we employ the same methodology as in the first step. The main difference here is that we consider the states corresponding to rigid body and flexible aircraft dynamics, and the updated states from the previous step is not included. In the third step of the algorithm, the system-relevant matrices A, B, C, and D are directly corrected using the updated states from the previous two steps.

The first error minimization problem which represents an output residual formulation between flight test data and model predictions by using only the output equations, is defined as:

$$\min_{\Delta x_{k,i}} \sum_{i=1}^{N_{\text{test}}} \sum_{k=1}^{N_t-1} \|\overline{y}_{k,i} - \mathbf{S}_{\text{sens}} \cdot (\mathbf{C}_\text{d} \Delta x_{k,i} + \mathbf{D}_\text{d}(\overline{u}_{k,i} - u_{0,i}) + y_{0,i})\|^2 \,. \tag{70}$$

We shall denote the value of perturbed state vector Δx_k that minimizes 70 by $\Delta x_\text{k}^{(c)}$, for fixed $\mathbf{C}_\text{d}$ and $\mathbf{D}_\text{d}$:

$$\Delta x_{k,i}^{(c)} \;=\; \arg\min_{\Delta x_{k,i}} \sum_{i=1}^{N_{\text{test}}} \sum_{k=1}^{N_t-1} \|\overline{y}_{k,i} - \mathbf{S}_{\text{sens}} \cdot (\mathbf{C}_\text{d} \Delta x_{k,i} + \mathbf{D}_\text{d}(\overline{u}_{k,i} - u_{0,i}) + y_{0,i})\|^2 \,, \tag{71}$$

where $\overline{u}_{k,i} \in \mathbb{R}^m$ and $\overline{y}_{k,i} \in \mathbb{R}^{\overline{l}}$ are measured inputs and outputs from ith flight test. N_{test} denotes the number of flight test sets used for the model updating method. For clarification: $\overline{l}$ is the number of outputs from test and l is the number of outputs from numerical model. $\mathbf{S}_{\text{sens}} \in \mathbb{R}^{\overline{l} \times l}$ is the sensor matrix allocating the measured outputs with the outputs from model. In most cases the use of a sensor matrix is necessary when you are interested in outputs which are essential but cannot be measured during the test or only a subset of measured quantities are intentional for model updating. The error minimization formulation given in 70 is used for the first and second step of the updating algorithm, where for fixed $\mathbf{C}_\text{d}$ and $\mathbf{D}_\text{d}$ the perturbed states Δx_k can be corrected.

The second error minimization formulation requires both state and output equations, where system matrices $\mathbf{A}_\text{d}$, $\mathbf{B}_\text{d}$, $\mathbf{C}_\text{d}$ and $\mathbf{D}_\text{d}$ can be corrected by using the updated states $\Delta x_\text{k}^{(c)}$ obtained from the Eq. 71, [129]:

$$\begin{aligned}\mathbf{A}_\text{d}^{(c)}, \mathbf{B}_\text{d}^{(c)}, \mathbf{C}_\text{d}^{(c)}, \mathbf{D}_\text{d}^{(c)} \;=\; \arg \min_{\begin{pmatrix} A_d & B_d \\ C_d & D_d \end{pmatrix}} \sum_{i=1}^{N_{\text{test}}} \sum_{k=0}^{N_t-2} \Bigg\| \begin{pmatrix} \Delta x_{k+1,i}^{(c)} \\ \overline{y}_{k,i} \end{pmatrix} - \\ + \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{S}_{\text{sens}} \end{bmatrix} \left(\begin{bmatrix} \mathbf{A}_\text{d} & \mathbf{B}_\text{d} \\ \mathbf{C}_\text{d} & \mathbf{D}_\text{d} \end{bmatrix} \begin{pmatrix} \Delta x_{k,i}^{(c)} \\ \overline{u}_{k,i} - u_{0,i} \end{pmatrix} + \begin{pmatrix} 0 \\ y_{0,i} \end{pmatrix} \right) \Bigg\|^2 .\end{aligned} \tag{72}$$

Here, the error minimization formulation in Eq. (72) is used in the third step of the updating algorithm. The following diagram (Fig.375) illustrates a summary of the study presented in this work, including the flight test domain, model structure, updating algorithm, and model validation procedure.

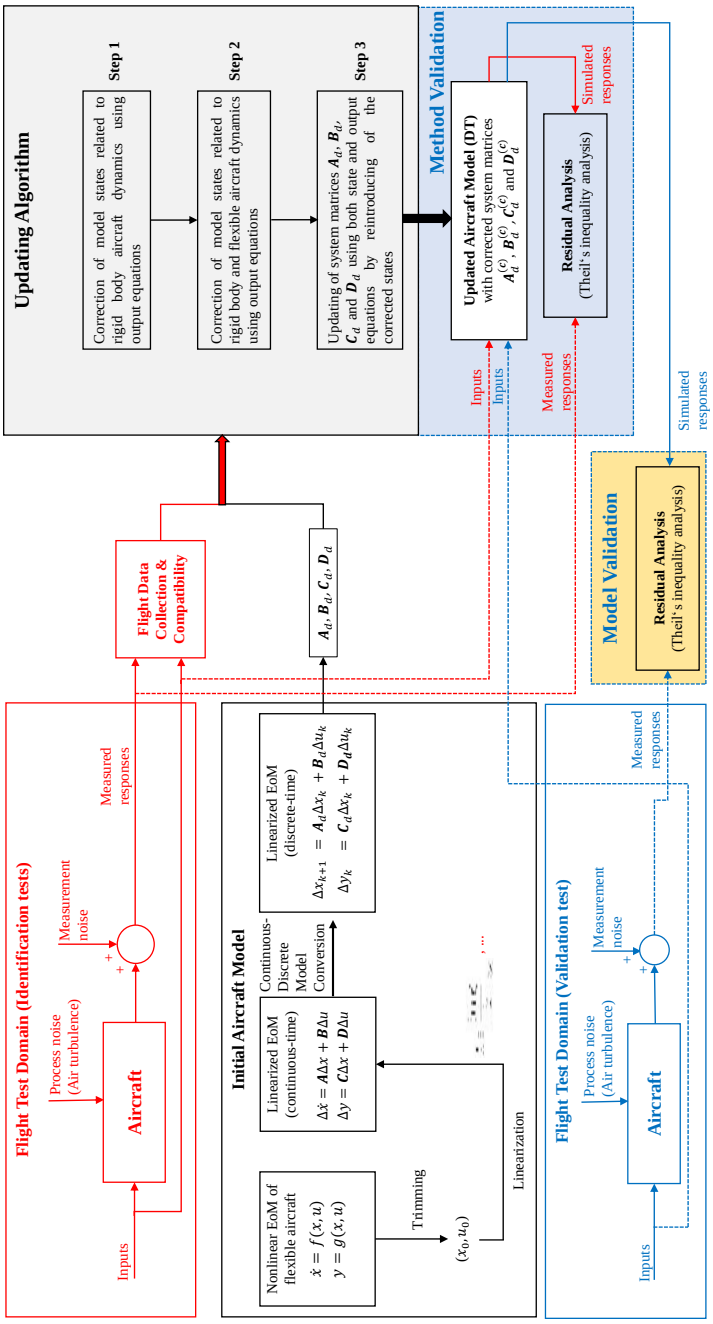


Figure 375: Overview of the flight test domain, model structure, updating algorithm and validation process [128]

Results

The FLiPASED aircraft is equipped with an integrated measurement system that is considered in both

the nonlinear aircraft model and the linearized state-space representation, respectively. The usual air data, position and inertial parameters are being logged on the aircraft. Attached to the front and rear spars are 12 inertial measurement units (IMU) that records the structural deflections of the wings. The wing-mounted IMUs measure translational accelerations in z direction and the angular rates ω_x and ω_y resolved in local coordinate system of the sensors. Further, an off-c.g.-mounted IMU at the fuselage provides flight measured translational accelerations $a_{\text{off-c.g.}} = [a_{x,\text{IMU-fuse}}, a_{y,\text{IMU-fuse}}, a_{z,\text{IMU-fuse}}]^T$ and the rotational rates $\Omega_{\text{off-c.g.}} = [p_{\text{IMU-fuse}}, q_{\text{IMU-fuse}}, r_{\text{IMU-fuse}}]^T$. A noseboom sensor provides the dynamic pressure, the altitude, the indicated airspeed $V_{\text{IAS,NB}}$, the angle of attack α_{NB} and the angle of sideslip β_{NB}. Figure 376 shows the configuration of the IMUs' placement on the FLiPASED aircraft [131]. The measurement coordinate systems of the sensors are illustrated in Figure 377. More information on flight test instrumentation (FTI) is given in Ref. [120].

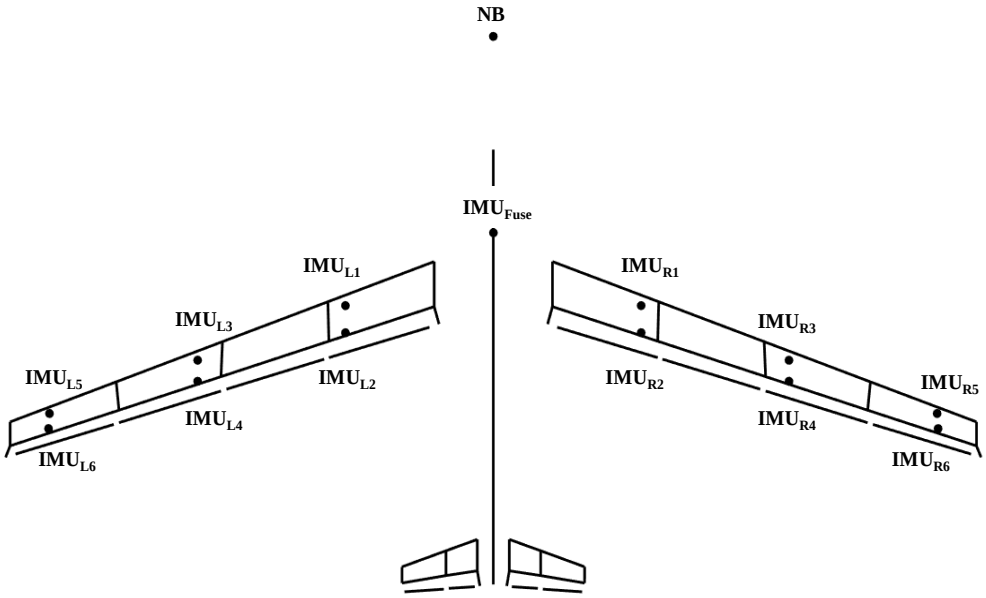


Figure 376: Sensor locations on FLiPASED aircraft

The flight test data used in this study are provided by a pushover-pull-up maneuvers. The primary objective of pushover–pull-ups, commonly known as roller-coaster, is to identify lift and drag characteristics, longitudinal stability, and elevator trim requirements. The maneuver starts from a trimmed level flight condition with a constant thrust [42]. With a sampling rate of 200 Hz and a 22-second time segment, experimental data from three test sets have been used for model updating method. The recorded input/output time series are then upsampled to 1 kHz to obtain data consistency with the discrete state-space model of the aircraft. All the flight measured outputs with their physical quantities can be found in [129], p.23. The trim conditions obtained from flight test measurements are stated in table below (Tab.82).

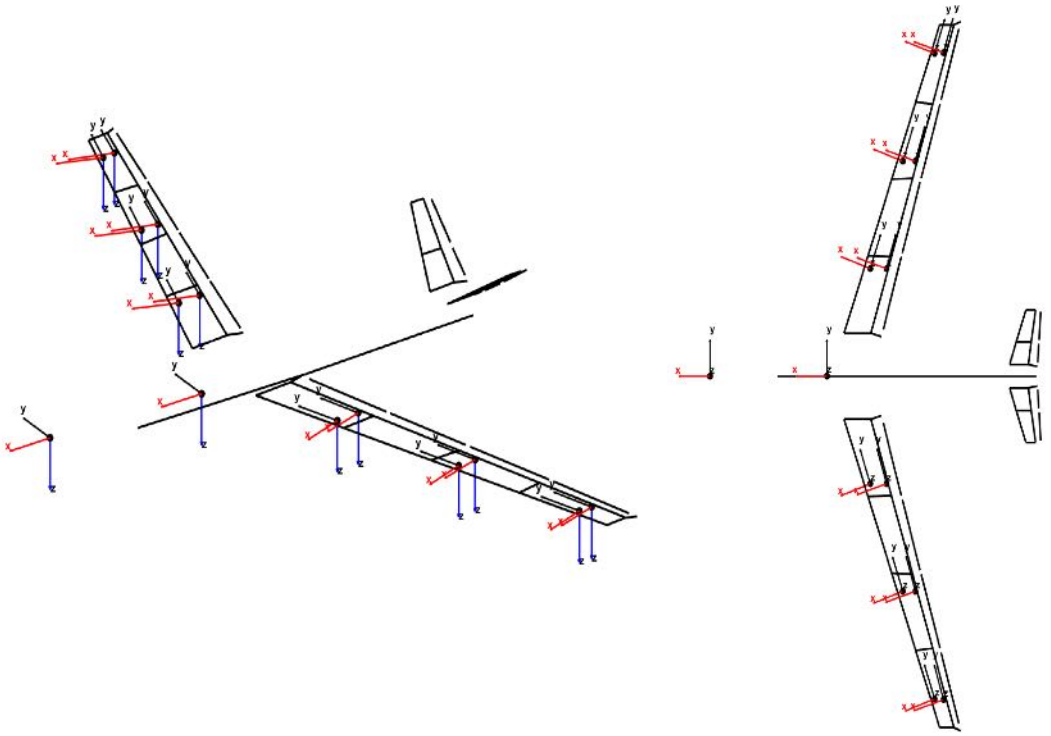


Figure 377: Measurement coordinate systems of the sensors on FLiPASED aircraft

Table 82: Trim values measured from identification flight tests

	θ_0 [deg]	$V_{\text{ias},0}$ [m/s]	$h_{\text{baro},0}$ [m]	$\alpha_{0,\text{NB}}$ [deg]
Identification Test #1	0.99	38.5	814	1.20
Identification Test #2	-1.0	38.8	773	5.03
Identification Test #3	0.41	43.2	719	1.43
Validation Test	1.79	39.7	738	3.12

The aircraft is excited for pushover-pull-up manoeuvres by elevator deflections, as depicted in Fig.378. Note, that in Fig. 378 the trim solutions are already subtracted from the elevator deflections.

The proposed updating method has been successfully applied on the linearized FLiPASED aircraft model including the use of the flight test data. Using residual analysis by means of Theil's inequality we assess the quality of the updated model for each of the output variables $o = 1, 2, ..., \bar{l}$. The formulation for Theil's inequality coefficient is defined as

$$\text{TIC}_\text{o} = \frac{\sqrt{\frac{1}{N_\text{t}}\sum_{k=0}^{N_\text{t}-1}\left(\overline{y}_{\text{k,o}} - y_{\text{k,o}}\right)^2}}{\sqrt{\frac{1}{N_\text{t}}\sum_{k=0}^{N_\text{t}-1}\left(\overline{y}_{\text{k,o}}\right)^2} + \sqrt{\frac{1}{N_\text{t}}\sum_{k=0}^{N_\text{t}-1}\left(y_{\text{k,o}}\right)^2}} \tag{73}$$

where TIC_o is a measure for fit error between measured and model outputs. Despite the fact that the acceptable value for TIC_o varies depending on the application, as a general rule, a value ¡ 0.25 indicates

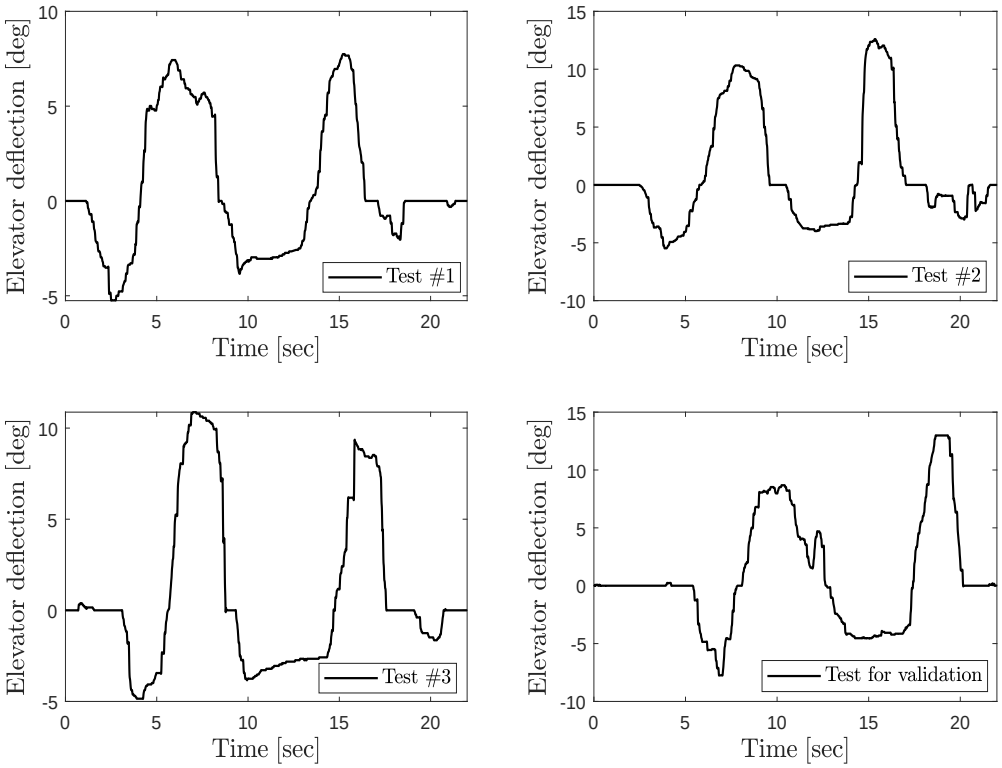


Figure 378: Elevator deflections used for pushover–pull-ups

an satisfactory agreement.

Figure 379 shows the fit error distribution results calculated from Theil's inequality formulation between the considered subset of output signals from three flight test sets and corresponding outputs from the initial and updated model.

As is evident from the plot in Fig.379, a high degree of fit between flight test data and results from updated linearized aircraft model has been achieved [129]. The mean Theil's inequality coefficient TIC_{mean} between the outputs from updated model and the recorded data from flight test is approximately 0.14 and has decreased by 0.16.

In the following, a few selected outputs from the flight test are plotted together with the outputs from initial and updated linearized model of the aircraft (Fig. 380 - 383). It is clearly evident that the presented updating method enables to reconstruct all the measured responses obtained from the flight test with high accuracy even in case of highly noise-contaminated experimental data [129].

For model validation within the framework of proof-of-match procedure, a flight test data set with the nearly identical control inputs and trim (initial) conditions is needed as chosen in identification tests for model updating [129]. Hence, a suitable set of test data is chosen for model validation, where the aircraft is excited for a pushover-pull-up manoeuvre by the elevator deflection shown in Fig.378. Results illustrated in Figure (385) demonstrate that a high degree of match has been achieved between outputs from flight test and outputs estimated from the updated aircraft model. The mean Theil's inequality coefficient TIC_{mean} between the outputs from updated model and the recorded data from the validation

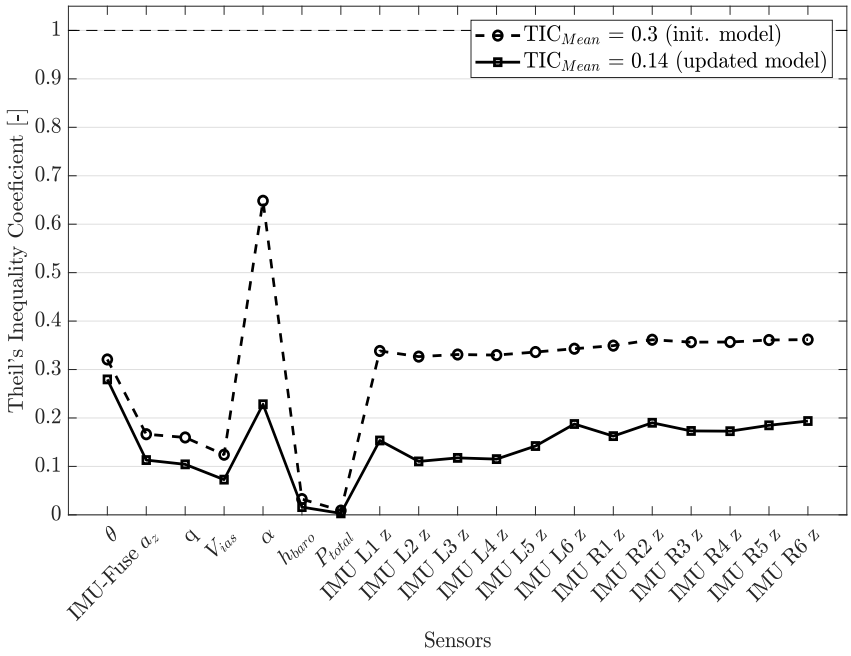


Figure 379: Fit error distribution between flight test and model (initial and updated) data for each output (Number of test sets = 3)

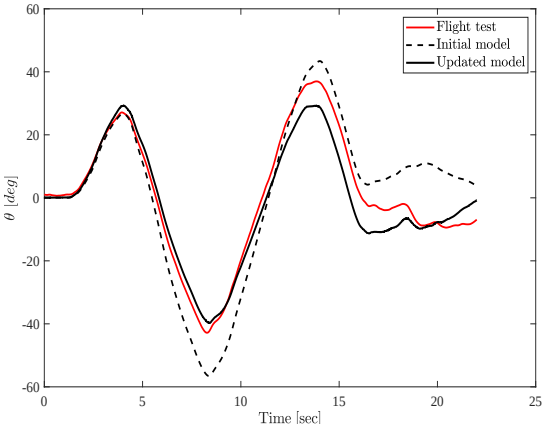


Figure 380: Pitch angle θ

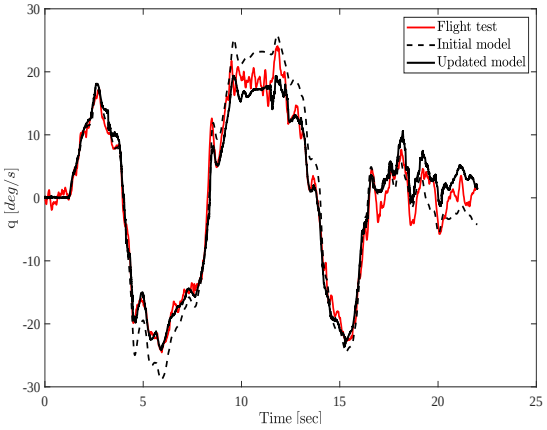


Figure 381: Pitch rate ($q_{\text{IMU-Fuse}}$)

test is approximately 0.14 and has decreased by 0.14. The diagram in Fig. 386 strongly indicates that nonsystematic error, represented by the covariance proportion, is the predominant source of fit error, indicating that the updated model is of high quality, i.e., it is able to replicate the true system response.

A detailed description of the updated method and its outcomes can be found in [129] and [128].

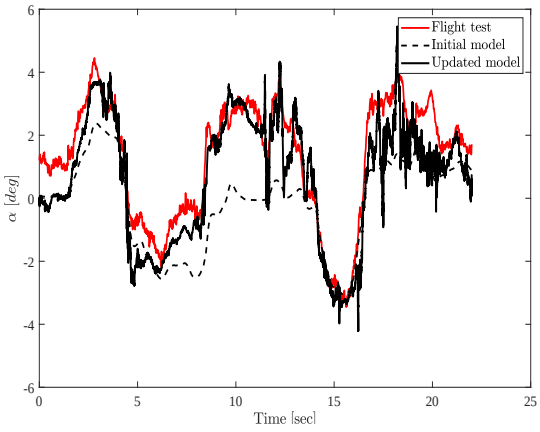


Figure 382: Angle of attack α

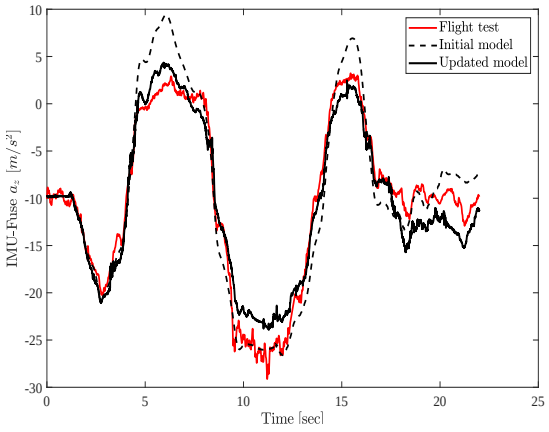


Figure 383: Vertical acceleration $a_{\text{zIMU-Fuse}}$

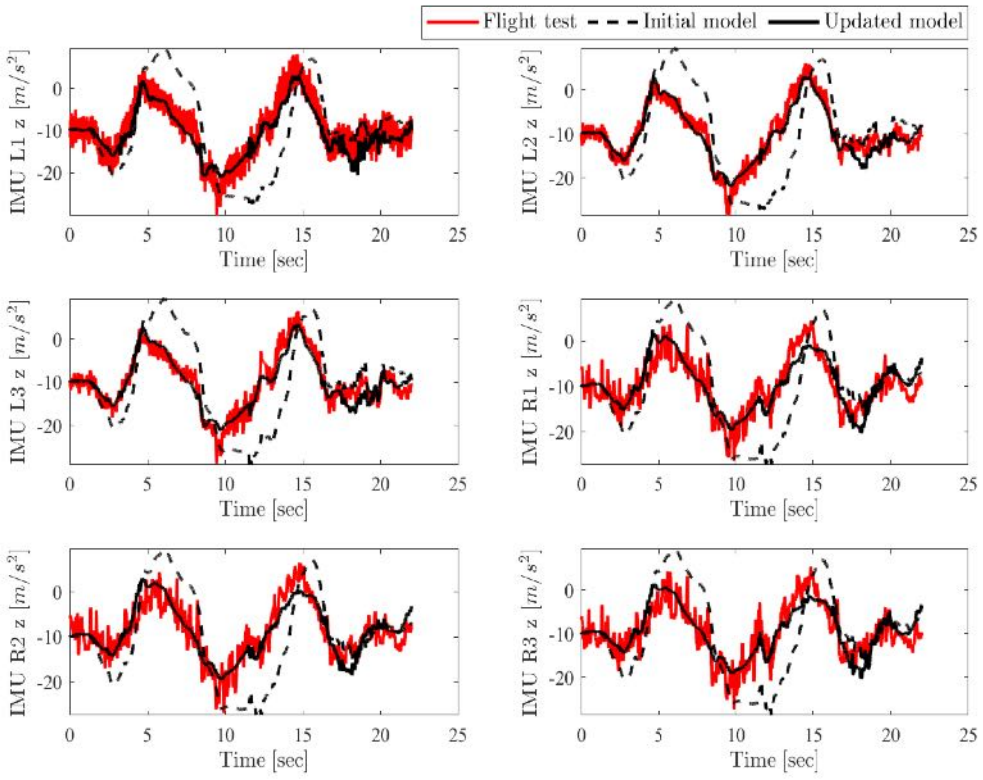


Figure 384: A subset of vertical accelerations $a_{z,\text{IMU}}$ recorded by six IMUs on the wings

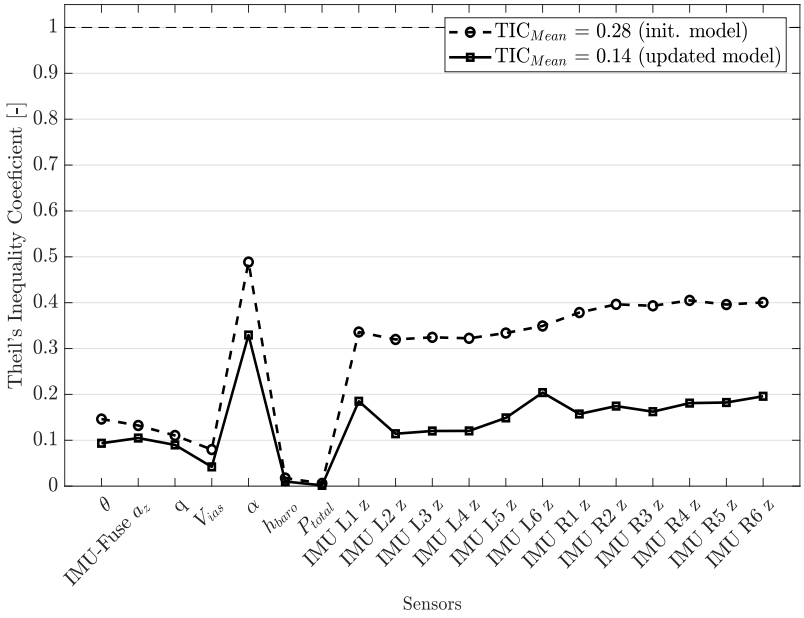


Figure 385: Fit error distribution between outputs from validation test and outputs provided from initial and updated model for proof-of-match procedure

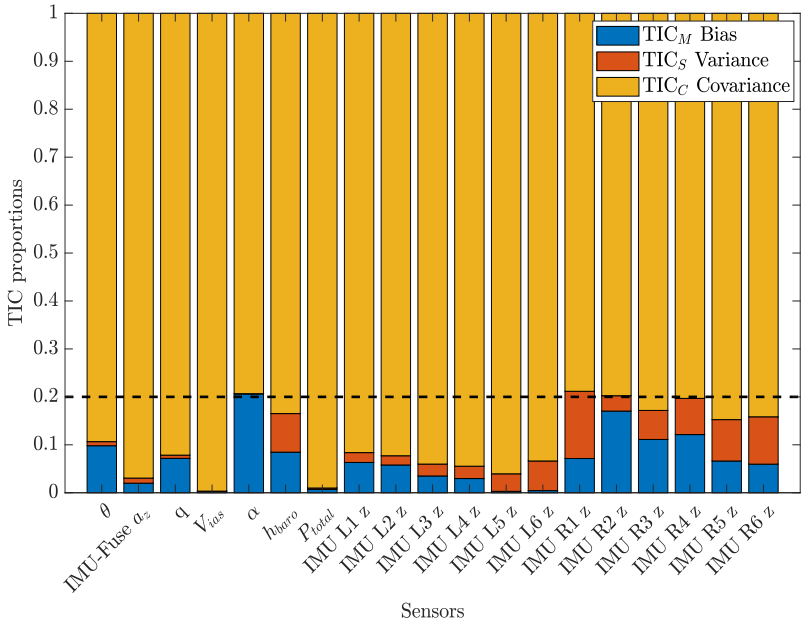


Figure 386: Breakdown of the fit error TIC$_o$ into proportions of bias, variance, and covariance (Updated Model vs. Validation test)

Model updating using modal parameters identified from GVT and open-loop flutter calculation (DLR, ONERA) For the synthesis of the flutter suppression controller and for a better prediction of the aircraft aeroelastic response and thus the onset of the open-loop flutter, the aeroservoelastic (ASE) model has been updated by incorporating modal parameters obtained from ground vibration tests. By using modal masses, damping ratios, eigenfrequencies and mode shapes identified from GVT, the structural dynamic model in modal domain has been generated and then directly integrated into the state-space formulation of the ASE model. This approach is appropriate if the intervention into the full FE model is not intended to update the structural dynamic model.

The open-loop flutter calculation with the updated model has been performed by means of *pk*-method. At the Mach number of 0.15 the predicted flutter velocity is $56,71$ m/s, and the flutter frequency is $8,65$ Hz as illustrated on the left side of the figure 387. The flutter mode involves a strong coupling between the first symmetric wing bending and the first symmetric wing torsion mode (critical modes) whereas the first symmetric wing torsion mode at $f = 10,7$ Hz dominates the motion at the flutter condition (Figure 387).

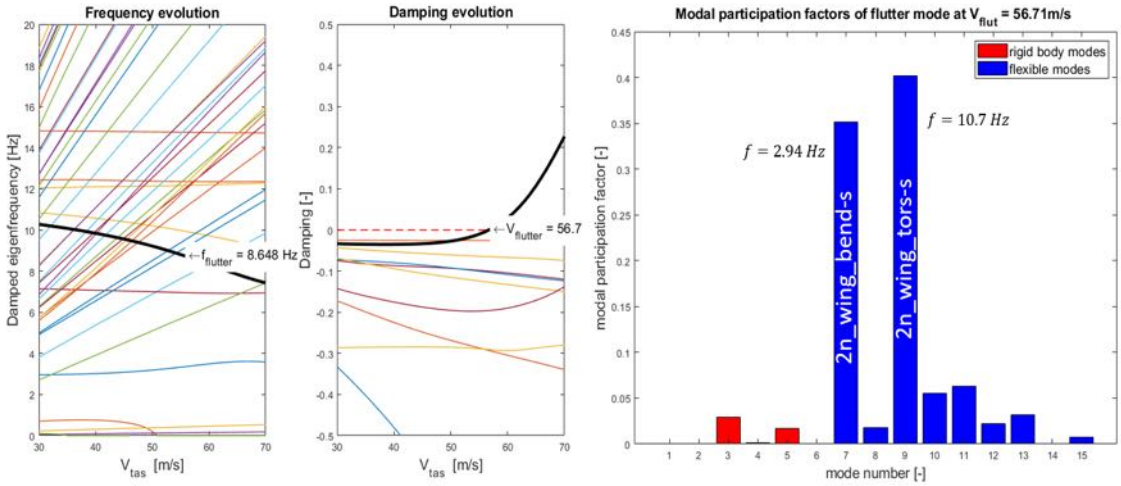


Figure 387: Computed open-loop flutter results (*pk*-method)

Data Compatibility Analysis

Data compatibility analysis was done to find the sensor errors after the first and second campaigns. It was assumed that the following sensor error model can be applied:

$$X_{measured} = k_X * X_{true}(t - \Delta t_X) + b_X \tag{74}$$

Here the $X_{measured}$ is the value as recorded by the instrument, k_X is the scaling factor for the X variable, X_{true} is the true value of the variable, t is the time step, Δt is the time delay for the recording and b_X is the bias for the variable. Here, k_X, Δt_X and b_X are kept constant for the flight (or flights).

After the first campaign where two flights were made, the following steps were taken:

1. Use the rudder doublet segment to estimate the β bias, scale factor, and time delay.
2. Fix the β error parameters.
3. Then use a segment with high change in α to estimate the α error parameters.
4. Then fix α parameters.
5. Go through the automatic injection manoeuvres while having α and β error parameters fixed. So only a_x, a_y, a_z and p, q, r were left free.
6. Correct each of the manoeuvre pair with the resulting sensor errors.

Table 83 notes the sensor errors for the aerodynamic angles.

Table 83: Sensor errors for flights FT24 and FT25.

Variable	Value	Error standard deviation
k_α	0.92655	0.014953
b_α	$0.033853 rad$	0.022071
k_β	0.99815	0.012045
b_β	$-0.012276 rad$	7.3434e-05
Δt_α	$0.066705 s$	2.5318e-05
Δt_β	$0.066802 s$	4.4859e-05

For variables a_x, a_y, a_z and p, q, r the scale factors and biases are slightly different for each manoeuvre.

Table 84 notes the preliminary sensor errors for flight tests FT27, FT28 and FT29.

Table 84: Sensor errors for flights FT27, FT28 and FT29.

Variable	Value	Error standard deviation
b_{a_x}	$0.0045802 m/s^2$	0.00046369
b_{a_y}	$-0.0030196 m/s^2$	0.00053258
b_{a_z}	$-0.0048884 m/s^2$	0.00011686
b_p	$0.00045561 rad/s$	1.1046e-06
b_q	$0.00099984 rad/s$	2.8646e-06
b_r	$-0.0016809 rad/s$	1.2129e-05
k_α	0.90503	0.002221
b_α	$0.017583 rad$	0.00010757
k_β	0.92837	0.0011371
b_β	$-0.01906 rad$	0.0001891
Δt_α	$0.093591 s$	0.0010538
Δt_β	$0.064484 s$	0.00056629

Flutter stopper tests (TUM) The flutter stopper mechanism's target was to ensure that the weight is shifted within the rod faster than two oscillations of the wingtip during flutter. At that time, the expected flutter frequency was around $8 Hz$, which meant a period of $0.125 s$. Therefore, the target actuation time was below $0.25 s$.

Figure 388 displays the wingtip movement of the left and right wing during the triggering of the flutter stopper in FT29. The release and capture points are well visible within the accelerometer measurements. The recorded time for the weight to change aft to fore was $0.120 s$ for the left wing and $0.115 s$ for the right wing. Longitudinal wing oscillations are visible for around $3 s$ after capturing the weight.

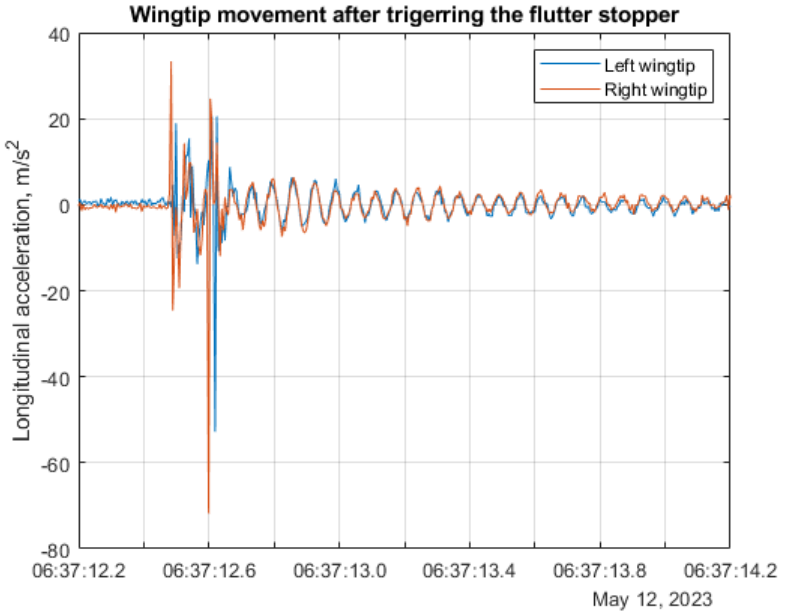


Figure 388: Wing oscillations after triggering the flutter stopper, FT29.

Open-loop flutter test points (sub-critical) (TUM, DLR, SZTAKI) In order to verify whether or not the flutter coupling was indeed active in the designed velocity range open loop flights were performed.

These flights were performed using constant bank angle circles with the autopilot keeping the speed as constant as possible. This was important to measure enough data of the aircraft in each linear time invariant system state. The measured MEMS acceleration data from FT32 is shown in Figure 389. The constant flight speeds were performed at 44 m/s, 46 m/s, 48 m/s, 50 m/s, 52 m/s, 53 m/s and 54 m/s. The velocity control worked well with variations rarely exceeding +-1 m/s. Since the aeroelastic flutter simulations which were updated with the 2023 Ground Vibration Test (GVT) results predicted a flutter speed of 56 m/s, it was decided to not fly faster than 54 m/s in open loop.

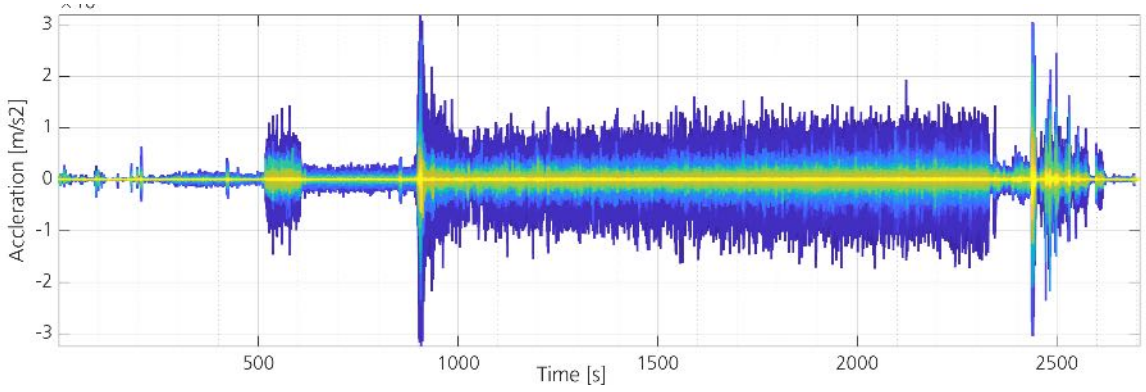


Figure 389: Measured MEMS acceleration data from FT32.

The result of a spectrogram using the P-Welch method can be seen in Figure 390. At approximately 1500 seconds the aircraft reaches 50 m/s. Here it can be seen that a mode at 10 Hz becomes dominant and starts to move down in frequency with increasing flight velocity. This is a first good indicator since the symmetric and anti-symmetric torsion modes were measured at 10 Hz and are expected to move down in frequency, ultimately resulting in a coupling of symmetric wing torsion with 2n wing bending.

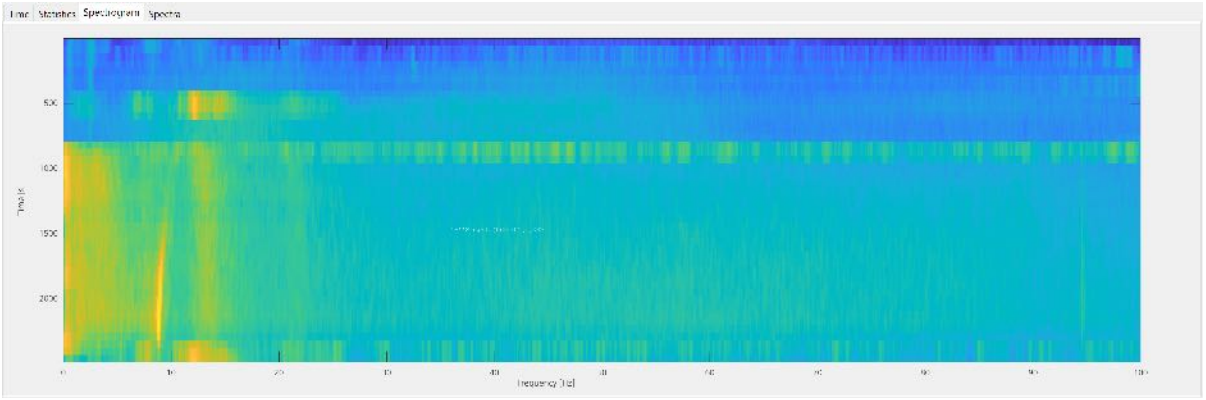


Figure 390: Spectrogram of a MEMS acceleration sensor from FT32.

The data is then analyzed with the Operational Modal Analysis (OMA) Online Monitoring (OLM) tool developed by DLR. The results of the Stochastic Subspace Identification (SSI) for frequency as a function of flight velocity can be seen in Figure 391 and damping as a function of flight veleocity in Figure 392. Furthermore the uncertainties are plotted as the shaded area around the system identification results.

By zooming on the flutter critical torsion mode in Figure 393 it can be seen that the symmetric wing torsion mode goes from 9.7 Hz at 44 m/s to 8.8 Hz at 54 m/s. Importantly the damping is seen to reduce from 3.3 % at 44 m/s to 0.4 % at 54 m/s. This result was obtained during the flight test and

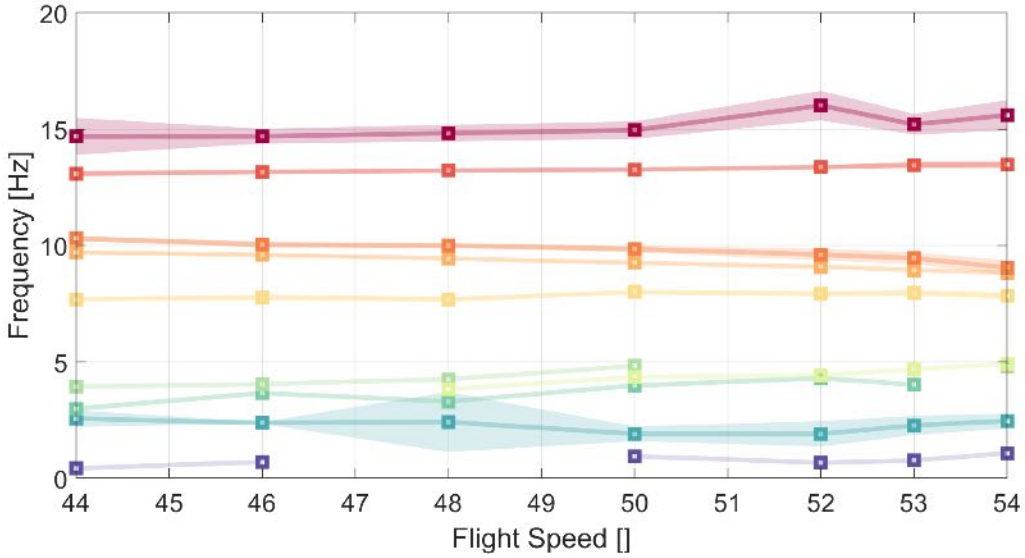


Figure 391: Frequency as a function of flight velocity from FT32.

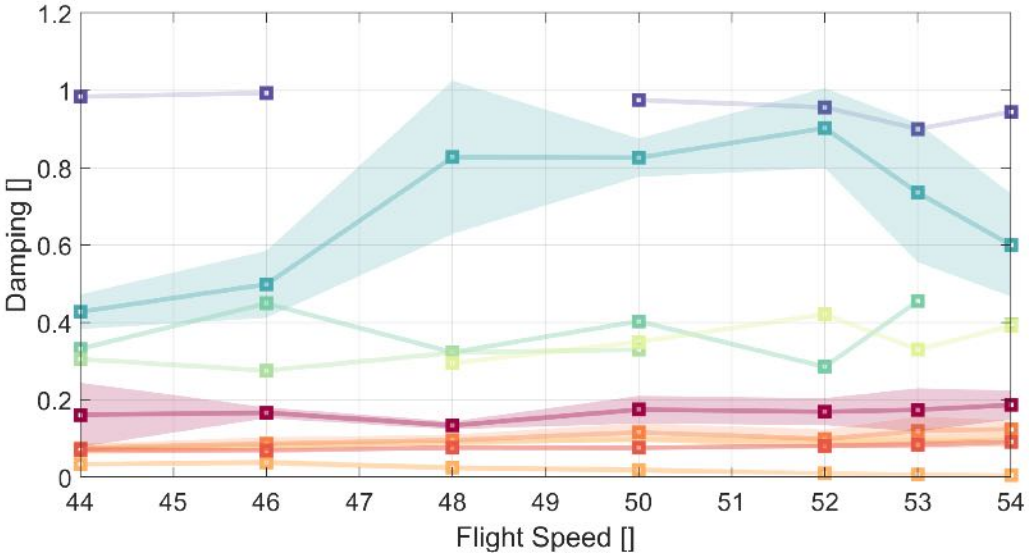


Figure 392: Damping as a function of flight velocity from FT32.

indicated that at 54 m/s the aircraft is right on the boundary of the flight stability. And that the aircraft was very close to a flutter coupling phenomenon.

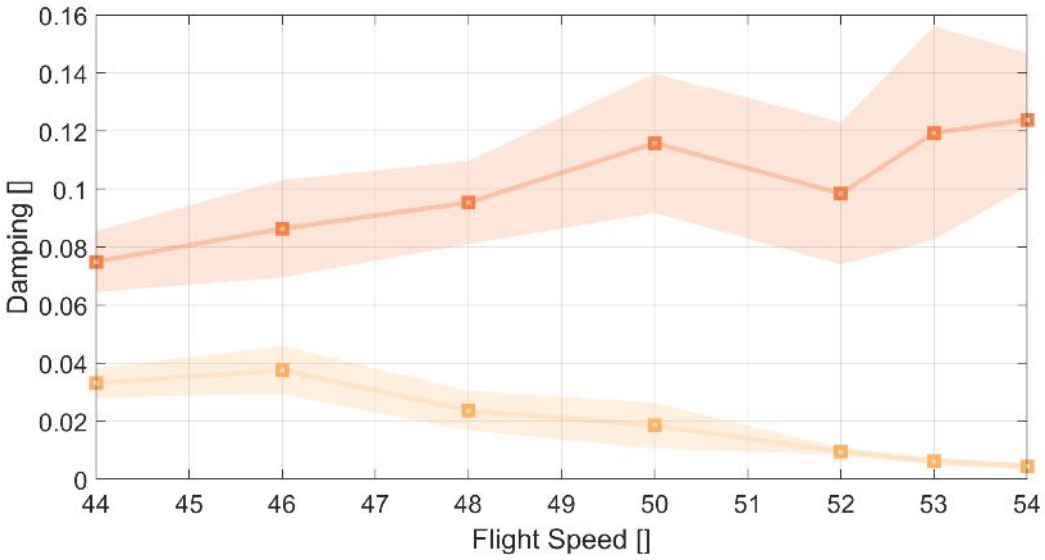


Figure 393: Damping of the wing torsion modes from FT32.

Closed-loop flutter test points (sub-critical) (TUM, DLR, SZTAKI) **SZTAKI controller test results**

The flutter suppression controllers employed in FT31, FT34, and FT36 were designed using the principles in [85]. The key points of the design are as follows. The flutter controller is synthesised with structured H_∞ synthesis with the objective to minimize the sensitivity function of the closed loop with minimal control effort. The generalized plant interconnection used for the design is depicted in Figure 394.

The controllers use two different set of sensor measurements. The first configuration (labelled ”4-input”) consists of p, q, q_L, and q_R, as illustrated in Figure 395. In addition, we also consider p_L, and q_R in the ”6-input” version. This case is also depicted in Figure 395. Two speed ranges are considered for the design: $[45,\ 58]\ \mathrm{m/s}$ and $[45,\ 61]\ \mathrm{m/s}$. In both cases, the end points and the middle point of the intervals are selected. Then, a single controller is synthesised to stabilize the system at these three speed values simultaneously.

Four controllers are designed in total: all combinations of four inputs, six inputs, $[45,\ 58]\ \mathrm{m/s}$ design range, and $[45,\ 61]\ \mathrm{m/s}$ design range. The results of the disk margin analysis in closed loop with these four controllers are depicted in Figure 396. The $[45,\ 61]\ \mathrm{m/s}$ versions do not provide sufficient margins around the open loop flutter speed, hence these did not make it to the flight tests. Based on Figure 396, the flutter controller is expected to expand the safe flight envelope by roughly 6%.

Mode	Open	Closed
2n wing bending	5.36 Hz 41.95 %	4.6 Hz 52 %
3n wing bending	7.94 Hz 11.63 %	7.94 Hz 16.18 %
Sym torsion	9.46 Hz 2.88 %	9.66 Hz 11.07 %
Anti torsion	9.94 Hz 9.77 %	10.92 Hz 24.6 %

Table 85: OMA flexible mode identification of DLR flutter controller at 48 m/s.

Mode	Open	Closed
2n wing bending	5.13 Hz 44 %	3.96 Hz 54.9 %
3n wing bending	7.89 Hz 12.6 %	8.56 Hz 23.7 %
Sym torsion	9.54 Hz 5.11 %	9.82 Hz 7.48 %
Anti torsion	10.09 Hz 14.2 %	10.04 Hz 20 %

Table 86: OMA flexible mode identification of SZTAKI flutter controller at 48 m/s.

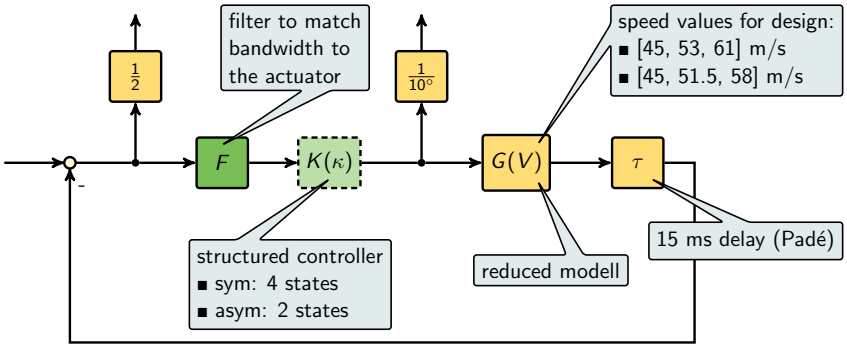


Figure 394: Generalized plant interconnection for the flutter controller synthesis.

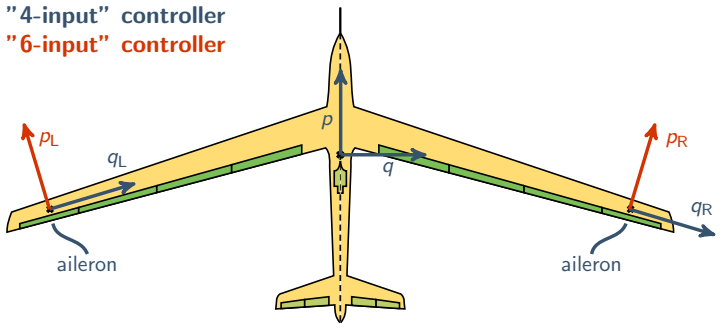


Figure 395: Sensor (and actuator) signals used for the flutter controller synthesis.

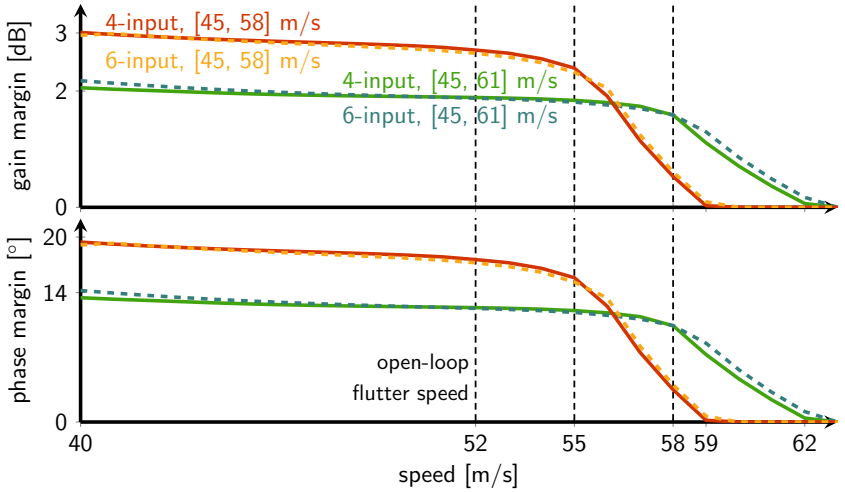


Figure 396: Gain and phase margin of the closed loop with the different versions of the flutter controller.

In the first flight test employing the ”6-input” controller (FT31), a dominant harmonic was found in the IMU signals close to 60 Hz. This phenomena is shown in Figures 397 and 398. The actuating signals, i.e. the aileron deflection commands are depicted in Figure 399. Based on this, the oscillations around 60 Hz were theorized to be caused by the flutter controller. In an attempt to suppress this peak, the controller was augmented with a notch filter in FT34 to make sure that it does not excite the system at 60 Hz. Also, the comparison between the actuator command and the measured aileron deflection in Figure 399 demonstrate that the direct drive has sufficient bandwidth for the flutter suppression task.

In FT34, two controllers were tested: the 4-input controller and the 6-input version augmented with a notch filter to suppress controller output at 60 Hz. Results with the 4-input controller are depicted in Figures 400, 401, and 402. The controller damps the vibrations at the flutter frequency (e.g. see Figure 400). However, the 60 Hz harmonic appear in the acceleration and angular rate signals.

Figures 400, and 401 show the results with the 6-input controller augmented with the notch filter. Based on these figures, the filter does not improve the high-frequency behaviour of the closed-loop, hence its use is deemed unnecessary. Those vibrations can be caused by the interaction of the structure of the wing with the direct drive actuator but not by the control law, as the test demonstrates.

The performance of the 6-input controller is slightly better then the 4-input version, as the local peaks in the IMU measurements are slightly less pronounced. Also, with the additional measured signals, the 6-input controller is expected to be more robust and hence safer to employ in the following high-speed flight test (FT36). Therefore, the 6-input controller without the notch filter was chosen for the high-speed flutter test.

DLR Active Flutter Suppression control synthesis, verification and validation

In this section the Active flutter suppression controller synthesis and validation for the UAV is detailed. A nonlinear aeroelastic model of the demonstrator aircraft is derived, which serves as a basis for flutter suppression controller design and validation. For controller design, the H2-optimal blending approach is applied, which allows isolating the critical flutter modes even if they are within the same frequency range. As a result, each isolated flutter mode can be stabilized by a separate Single Input Single Output (SISO) controller which is scheduled with airspeed. A detailed description of the design, synthesis and tuning of the overall flutter suppression controller will be detailed. The promising results of the achieved flight envelope expansion, validated by flight tests, are discussed in following sections.

Controller Design and Synthesis

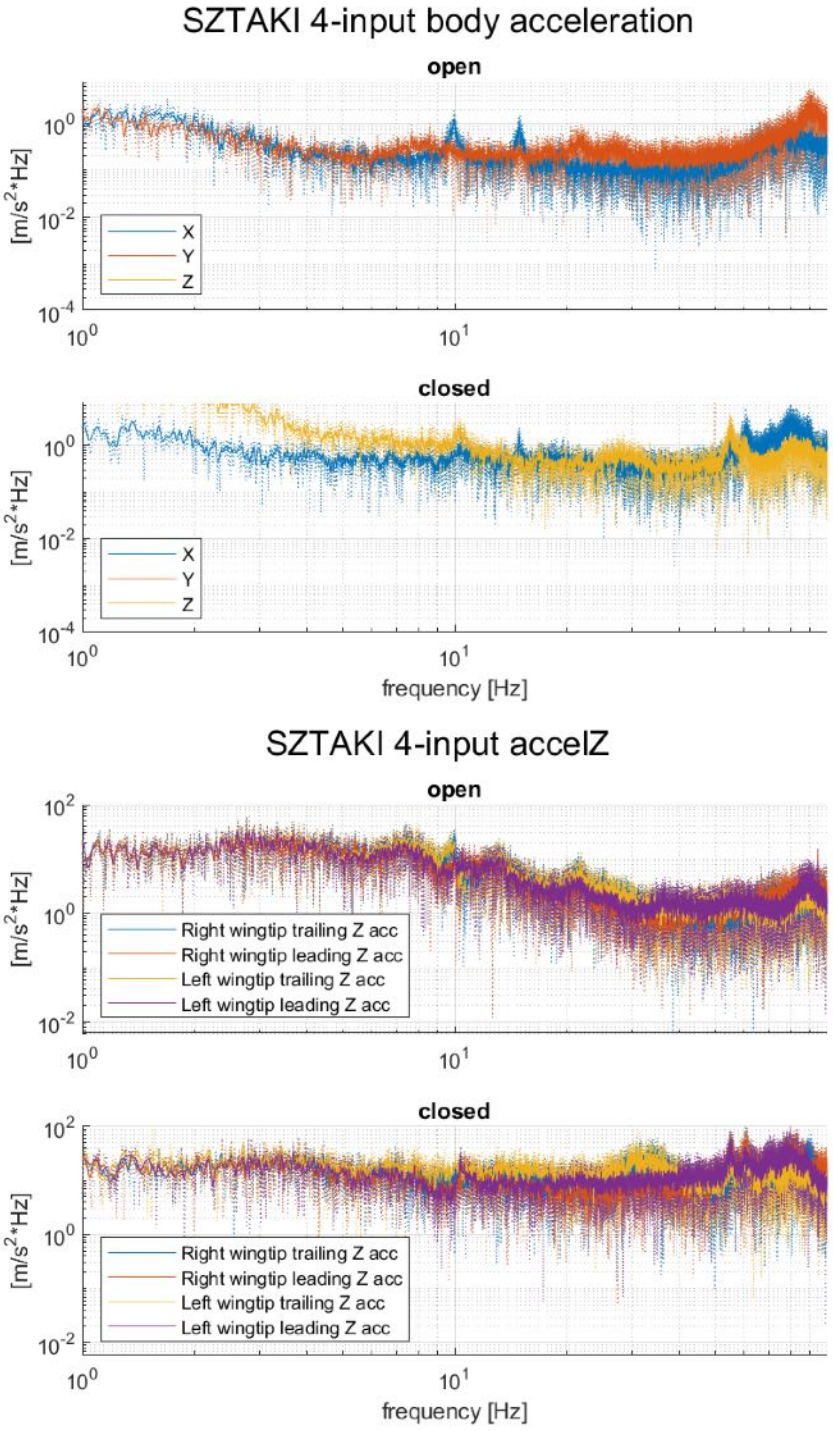


Figure 397: FT31 accelerations in the frequency domain.

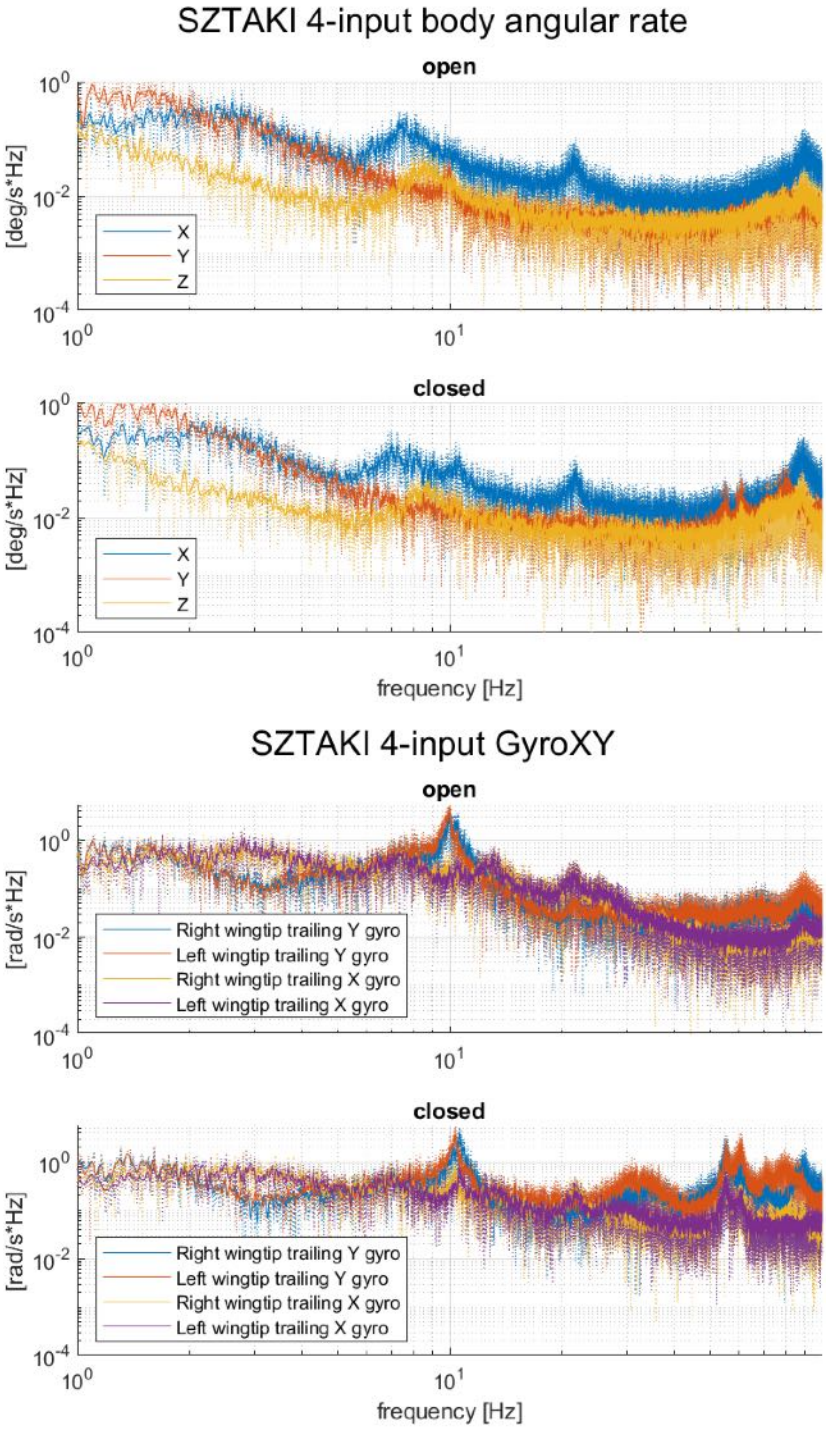


Figure 398: FT31 angular rates in the frequency domain.

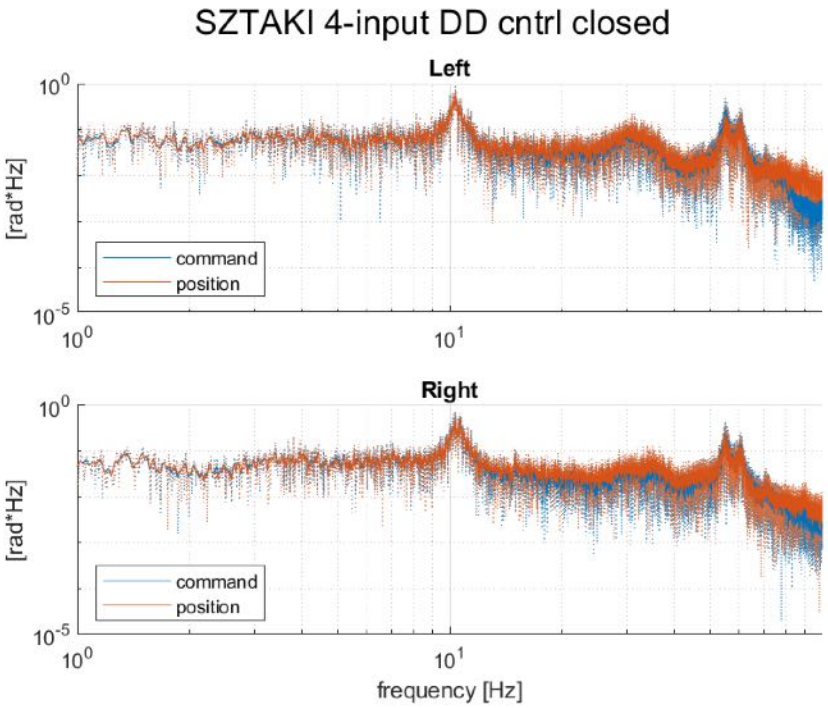


Figure 399: FT31 aileron command in the frequency domain.

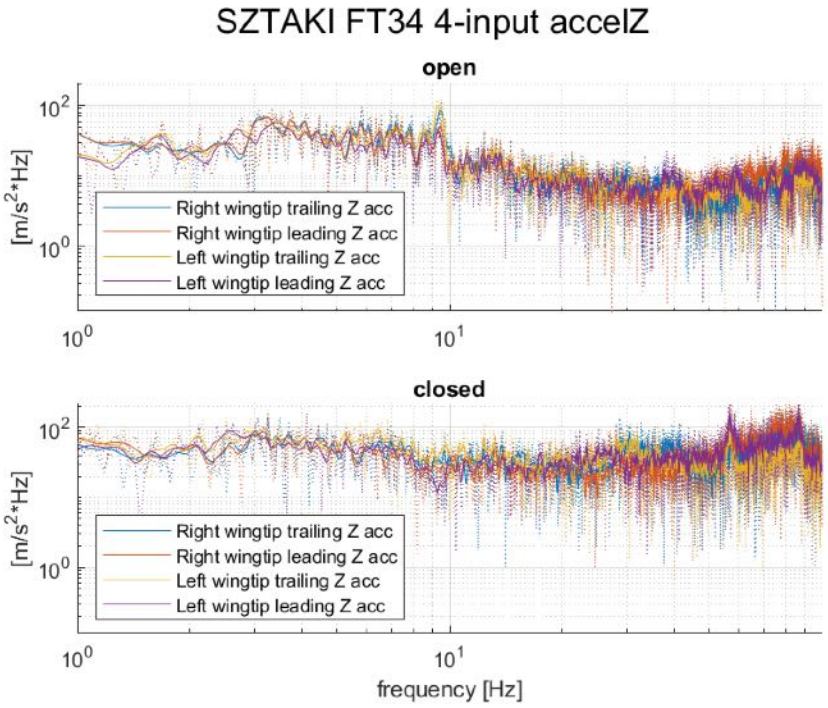


Figure 400: FT34 with the 4-input controller: accelerations in the frequency domain.

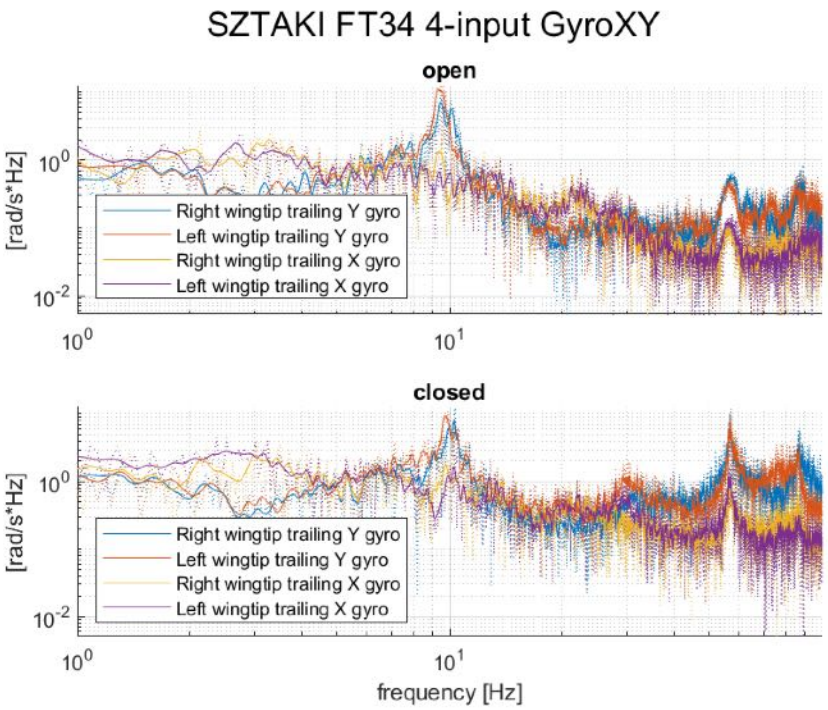


Figure 401: FT34 with the 4-input controller: angular rates in the frequency domain.

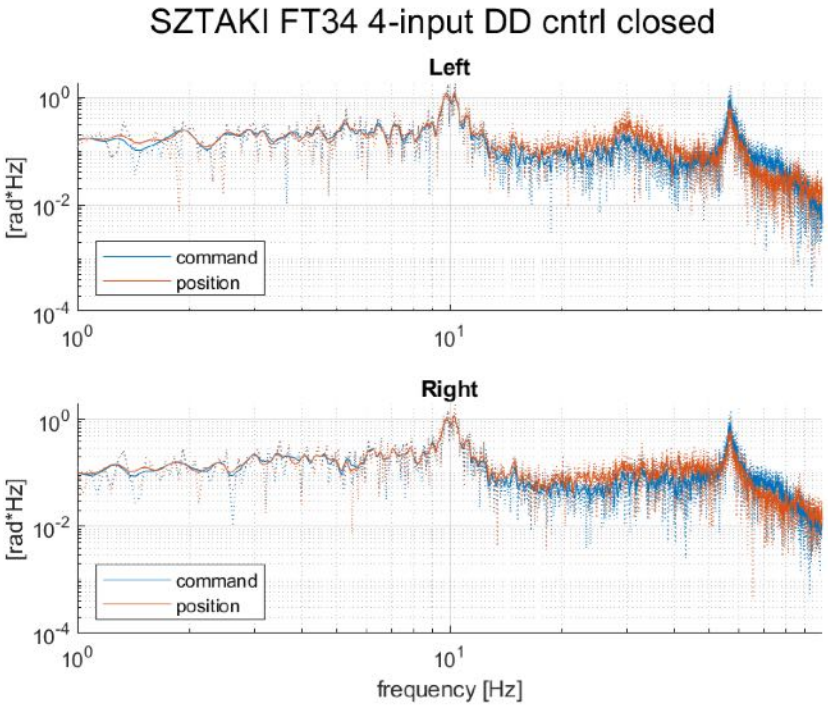


Figure 402: FT34 with the 4-input controller: aileron command in the frequency domain.

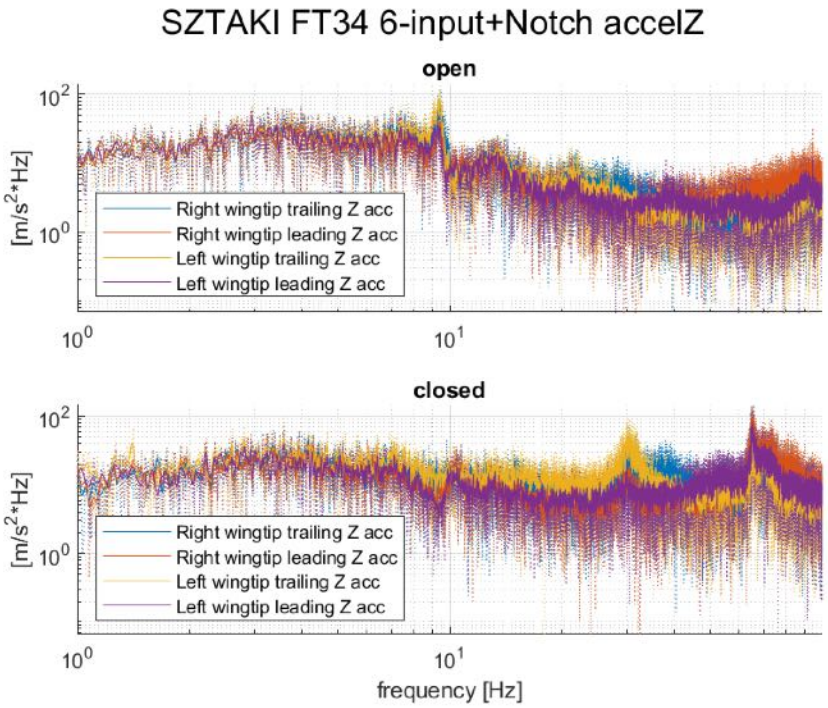


Figure 403: FT34 with the 6-input controller: accelerations in the frequency domain.

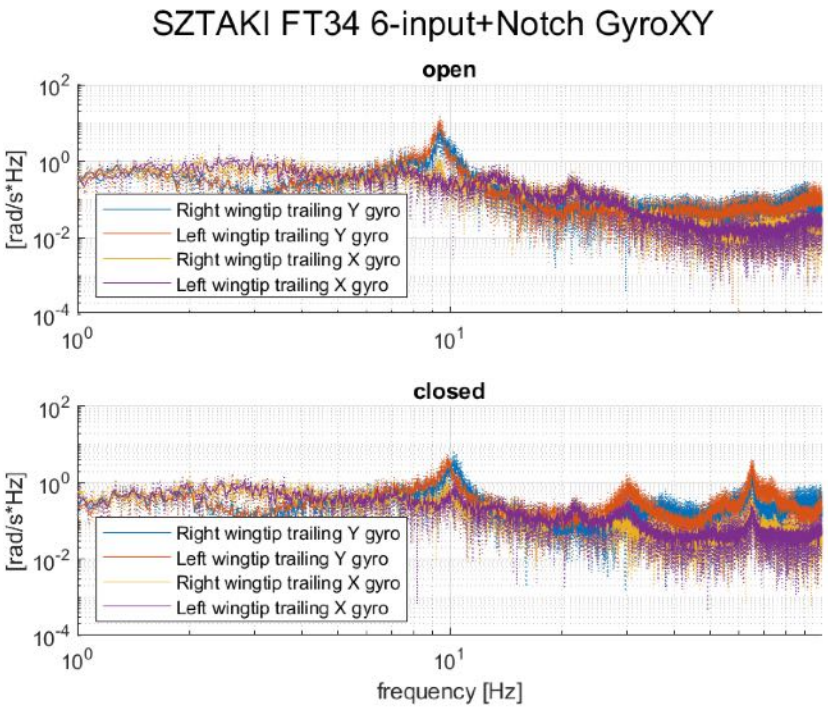


Figure 404: FT34 with the 6-input controller: angular rates in the frequency domain.

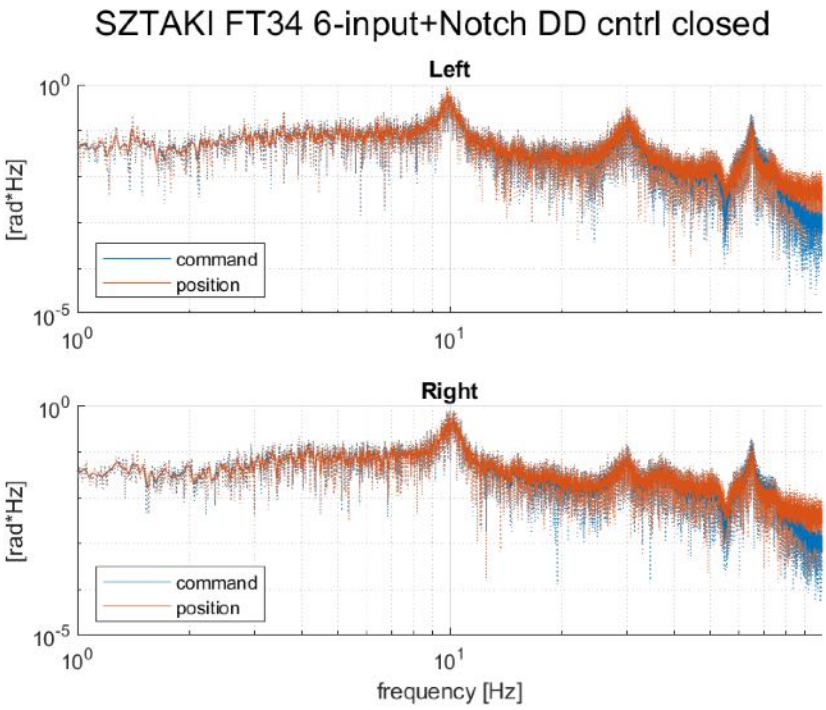


Figure 405: FT34 with the 6-input controller: aileron command in the frequency domain.

For the controller design, the nonlinear aeroelastic model of the UAV is linearized around steady horizontal flight at different airspeeds. Subsequently, a modal decomposition is carried out on each of the resulting LTI systems allowing for an individual analysis of the obtained aeroelastic modes. Both unstable modes describe flutter mechanisms based on a coupling of wing bending and torsion, where the mode shape is of symmetric and asymmetric nature, respectively. This is illustrated in a pole zero plot of the linearized aircraft model as shown in the figure 406. While the symmetric flutter mode becomes unstable at around 50 m/s with 8.0 Hz (50.3 rad/s), the asymmetric flutter mode follows at 52 m/s with 7.4 Hz (46 rad/s). To increase the aircraft's operational velocity range, it is hence required to stabilize the flutter modes, which is referred to as flutter suppression.

The two flutter modes, which are close in frequency but well distinguishable by their mode shapes, are decoupled and an individual mode stabilization by dedicated SISO controllers is enabled using H2-optimal blending approach. The control inputs considered for active flutter suppression are the deflections commanded to the outermost aileron (powered by a high bandwidth direct drive actuation system) on each wing. As measurement signals, the pitch rate and vertical acceleration captured by the 13 different IMUs are considered. Since the mode shapes change only slightly within the critical airspeed range, it is sufficient to compute the blending vectors at a single airspeed of 60 m/s and hold them constant within the whole flight envelope. With the derived blending vectors, it is possible to design dedicated SISO controllers for the symmetric and asymmetric flutter mode. Furthermore. The SISO controller is augmented with a band-pass filter to ensure that no interference occurs with the flight control system operating at lower frequencies and that higher frequent modes are not excited. Since a large velocity range needs to be considered, the core of the flutter suppression controller is gain-scheduled with indicated airspeed. As explicit optimization constraints, a gain margin of 6 dB and a phase margin of 45◦ are demanded on the blended input channel at an indicated airspeed of 30, 40, 50 and 60 m/s. The resulting constrained multi-model optimization problems are non-convex and solved using Matlab's systune routine based on non-smooth optimization techniques.

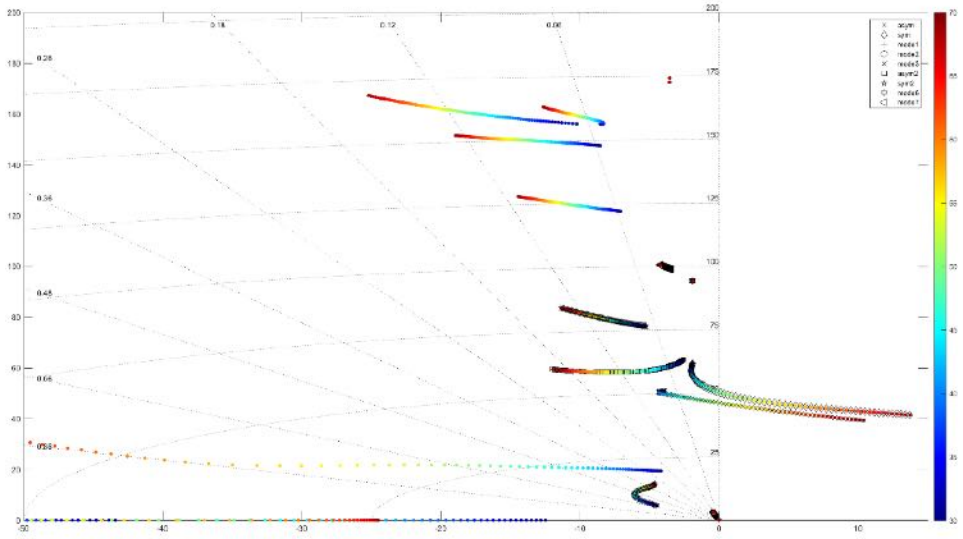


Figure 406: Open Loop Pole Zero Plot for DLR Flutter Control Synthesis

.

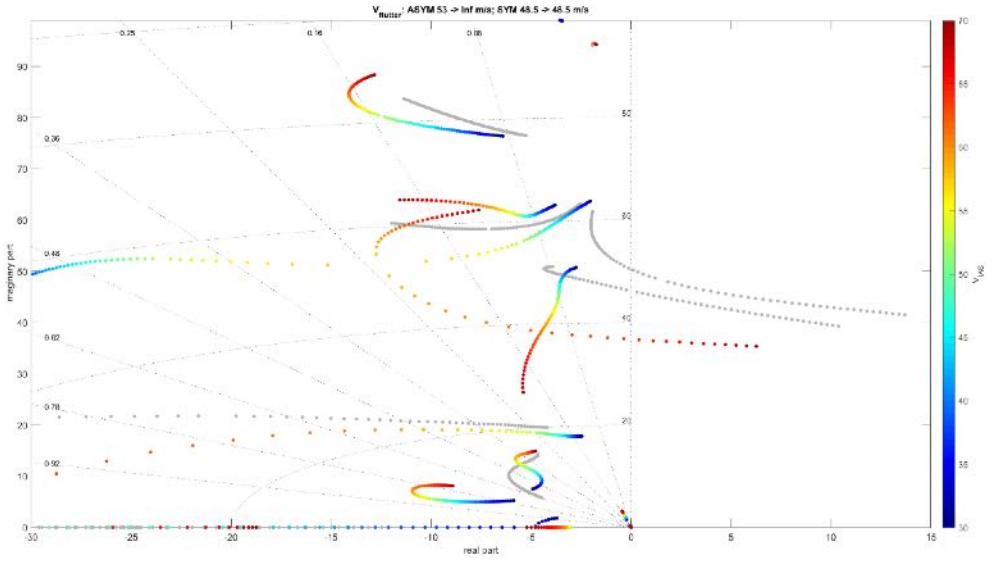


Figure 407: Closed Loop Pole Zero Plot with DLR Active Flutter Suppression Control

.

Closing the two SISO loops (symmetric and asymmetric) stabilizes both flutter modes as it is illustrated in the following pole migration plot, figure 407. The plot compares the open-loop poles in gray to the closed-loop poles depicted in color dependent on the airspeed. Clearly visible is the unstable behavior, i.e., the crossing to the right half plane of the first (symmetric) and second (asymmetric) flutter mode in the openloop. With the flutter suppression controller, the symmetric flutter mode can be stabilized up

to airspeeds of 62 m/s. The asymmetric mode is stabilized even beyond 70 m/s. This flight envelope expansion using Active Flutter Suppression system is visualized in the figure 408, where simulation results indicate we could now fly the demonstrator aircraft till 62 m/s, thus achieving around 11 m/s increase in flight envelope. Beyond this speed, one of the flutter modes become unstable, represented in the figure 408, as damping value of this critical mode becoming negative.

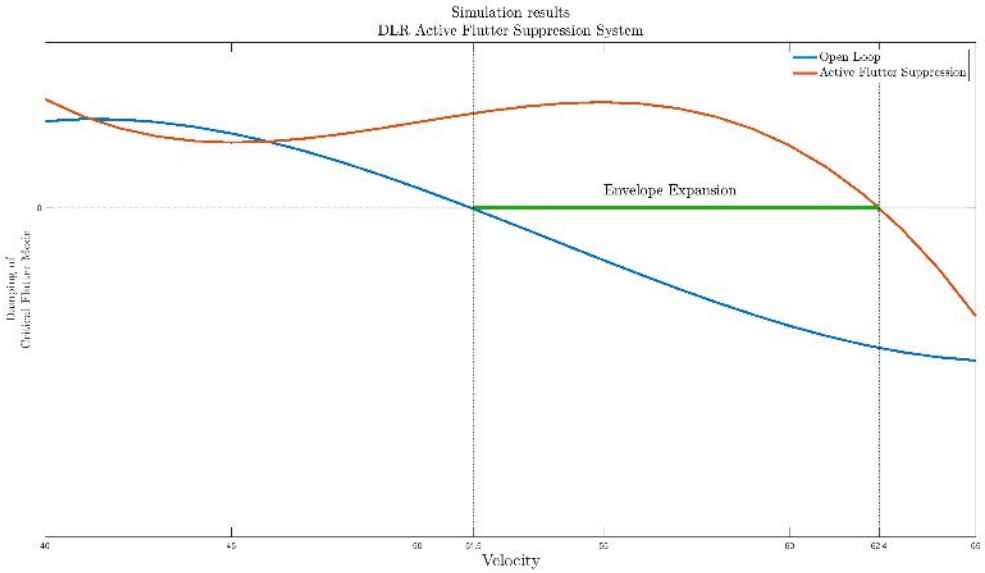


Figure 408: Flight Envelope Expansion with DLR Active Flutter Suppression Control

.

Flight Test Results(sub-critical) This section presents the experimental validation of the DLR Active Flutter Suppression System through Flight Tests 33 and 35. The objective of these tests was to assess the effectiveness of the system in suppressing flutter. Flight Test 33 involved subcritical flutter speeds and demonstrated the system's performance with the flutter controller turned on and off. Flight Test 35 was aimed to evaluate the system's behavior at critical flutter velocities. The test results were analyzed using time series data, Fast Fourier Transform (FFT) analysis, and comparisons of open and closed loop data.

The maiden flight for testing DLR active flutter suppression system is conducted in Flight test 33. The plan was to fly the demonstrator aircraft in sub critical flutter speeds and then switching on the Flutter controller. The objective of this flight test is to validate the effect of Flutter suppression System, which can be inferred through a relative increase in the damping of Flutter critical modes with Flutter controller ON as compared to without Active Flutter Suppression system, former referred to as Close Loop(CL) and the latter referred to as Open loop(OL) in the following plots. The velocity profile during this flight test and Direct Drive output i.e. Flap L4 and R4(Actuators used for Flutter Suppression) are shown in figure 409. A total of 3 velocity points are chosen viz,. 44m/s, 45m/s and 46m/s to evaluate the flutter controller behaviour and the demonstrator is flown with and without flutter controller at all these 3 test points. The recorded data from these tests points can be visualized in figure 409 represented as OL(Active Flutter Suppression OFF) and CL(Active Flutter Suppression ON). It can be seen that Flap L4 and R4 are active in Closed Loop and the activity of the flutter actuators is increasing with increasing velocity.

Figures 410 and 411 show the Active flutter suppression performance on flexible modes, where wing tip accelerations and angular rates are plotted along with flutter actuator commands of both left and right

wing. Note that DLR Active flutter suppression system uses z axis accelerations from all 13 IMU's(6 on left wing, 6 on right wing and 1 in fuselage) and y-axis angular rates from 7 IMU's (3 on left wing trailing edge, 3 on right wing trailing edge and 1 in fuselage). Here only measurements on wingtip IMU's of left and right wing are shown.

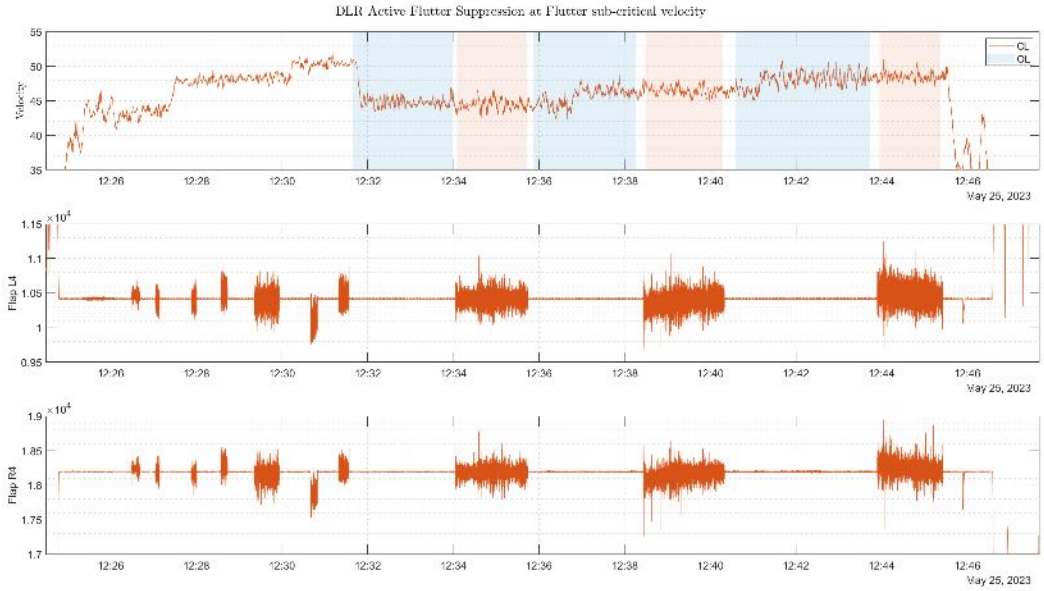


Figure 409: Flight Test Results(sub-critical) - Time Series Plot for FT 33

.

As it is difficult to evaluate the damping behaviour of flutter controller using time series data, FFT analysis on wing tip sensor data is conducted and the FFT results comparing Open and Closed loop data at 46 m/s test point is shown in figure 412. The analysis clearly indicates that the flutter controller effectively reduces the magnitude of flexible modes around flutter frequencies (e.g., 8Hz). Note that we are still flying in sub-critical flutter velocity and the damping is only expected to increase as we get close to flutter critical speeds.

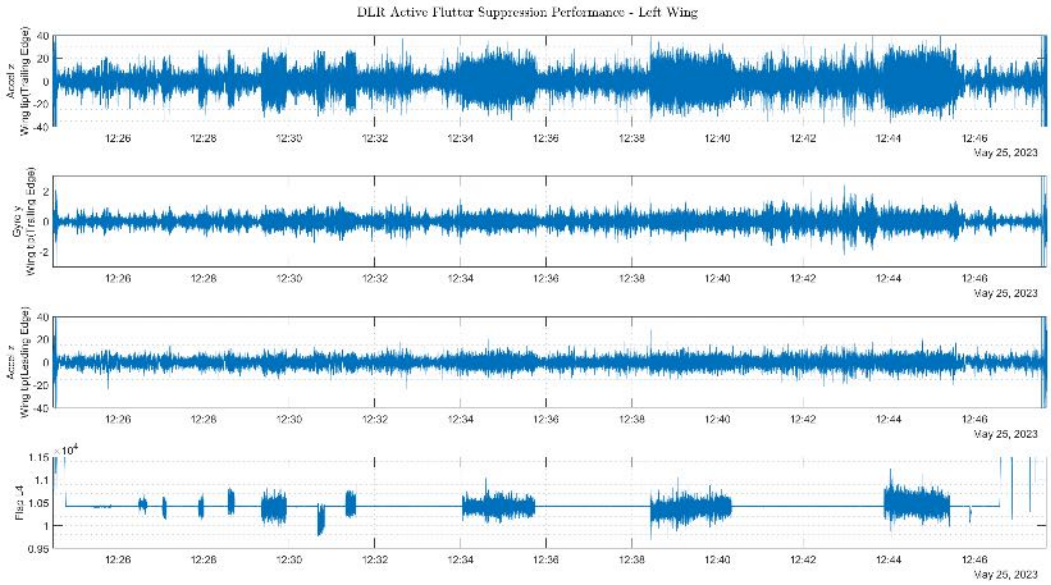


Figure 410: Flight Test Results(sub-critical) - Time Series Plot for FT 33

.

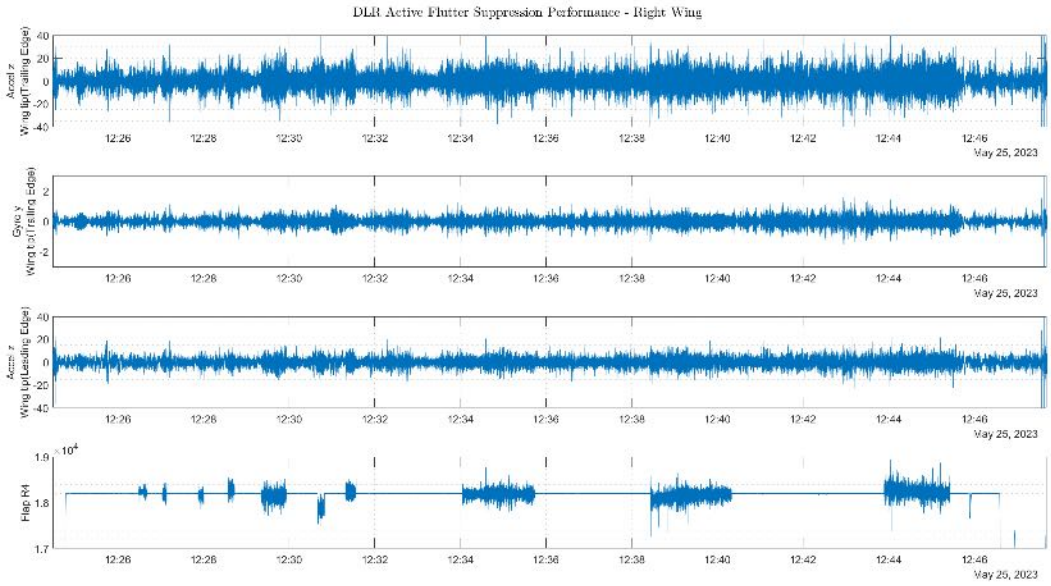


Figure 411: Flight Test Data(sub-critical) - Time Series Plot for FT 33

.

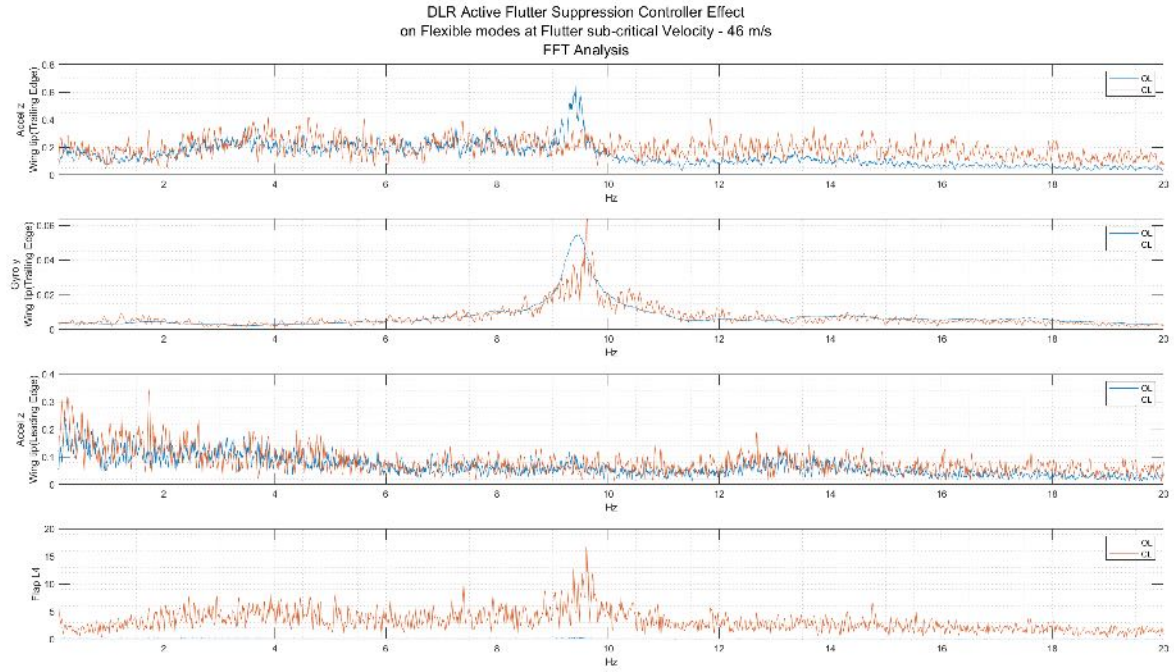


Figure 412: Flight Test Results(sub-critical) - FFT Analysis

.

Closed-loop flutter test points (critical) (TUM, DLR, SZTAKI) DLR controller test results

Based on the successful results obtained from Flight Test 33 with DLR Active Flutter Suppression System, Flight Test 35 was aimed to evaluate the system's performance at critical flutter velocities. The time series plot of the velocity profile and the instant at which flutter controller is active is shown in figure 413. At 54m/s it was decided to switch off the flutter controller to compare the open loop and closed data, as it was predicted that flutter phenomenon is expected around this velocity(which was later confirmed from Open Loop tests in Flight test 37). Beyond this flutter critical velocity, it was decided to fly only with Active Flutter suppression system ON. We can see that demonstrator could successfully fly beyond the flutter velocity and a maximum velocity of 59m/s(peaks of 61m/s) was achieved, while flutter controller is actively suppressing the flutter. This flight envelope expansion from flight test results is visualized from figure 414. Thus, a 5m/s flight envelope expansion is achieved through this Active flutter suppression system. It is noteworthy to mention that, as we didn't fly beyond 59m/s with active flutter suppression ON, the limit to the envelope expansion is at-least 59m/s, with simulation results indicating that we could have achieved even higher velocities. It is important to note that the flutter critical velocity is 4m/s later than that was predicted from available models(as shown in figure 408). The decision to not fly beyond 59m/s is motivated by the fact that we needed to verify open loop flutter velocity in subsequent flight tests.

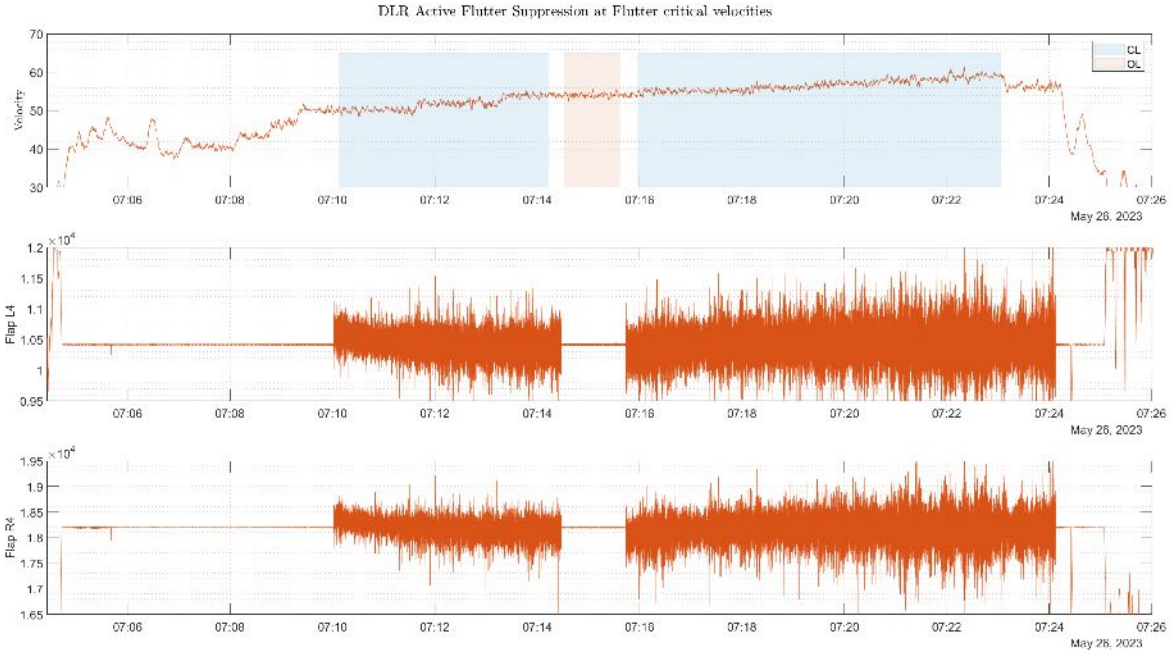


Figure 413: Flight Test Results(critical)- Time Series Plot for FT 35

.

The DLR Active Flutter Suppression system can be visualized from time series plots as shown in figures 415 and 416.

The test point of 54m/s wherem the instant at which flutter suppression system is switched off, is plotted in figure 416. We can clearly see that as soon flutter controller is switched off, the flexible modes start appearing with tendency of increasing magnitude, inferred from the wing tip sensor data.

To further verify the performance of flutter suppression system, FFT Analysis of wing tip sensor data

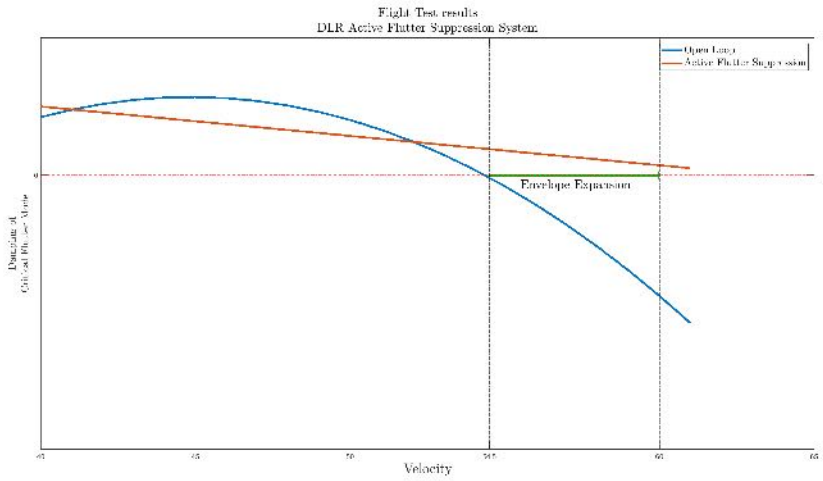


Figure 414: Validation of Flight Envelope Expansion with DLR Active Flutter Suppression Control

.

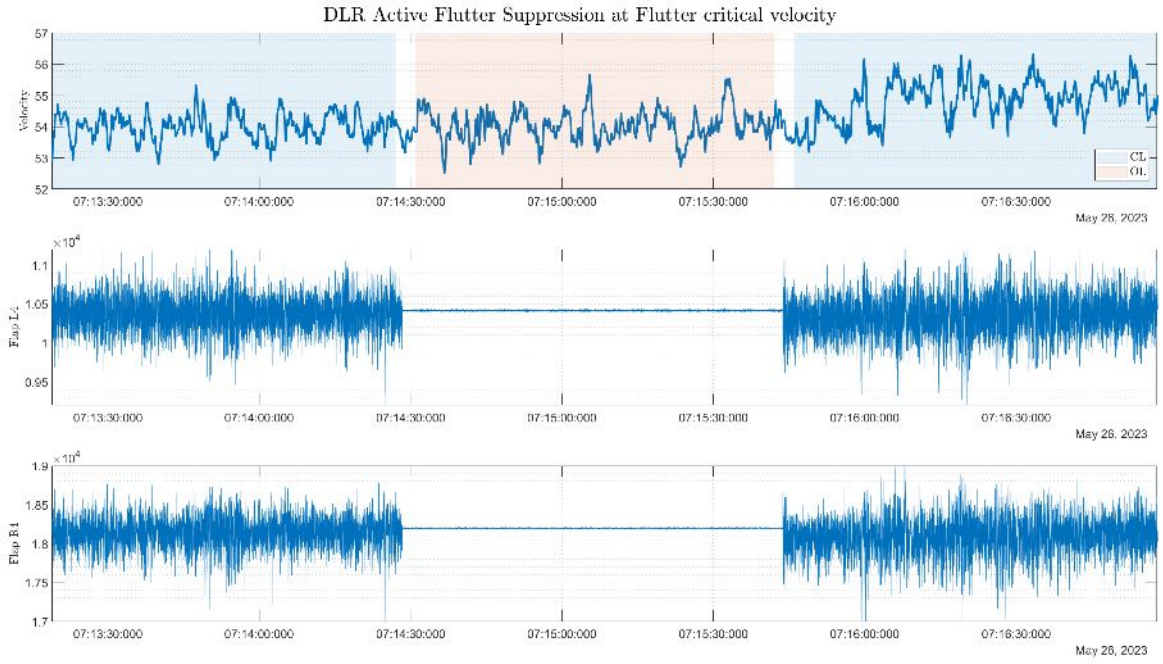


Figure 415: Flight Test Results(critical) - Time Series Plot for FT 35

.

is conducted as shown in figure 417. The damping effect of flutter suppression system is significant, as the magnitudes of flexible modes are greatly reduced and this relative increase in damping between open and closed loop is more prominent at critical flutter velocities(compared to sub-critical velocities as shown in figure 412). This improvement in damping at higher velocities is expected.

Furthermore, to verify good separation of rigid body controller with flutter controller, FFT Analysis of

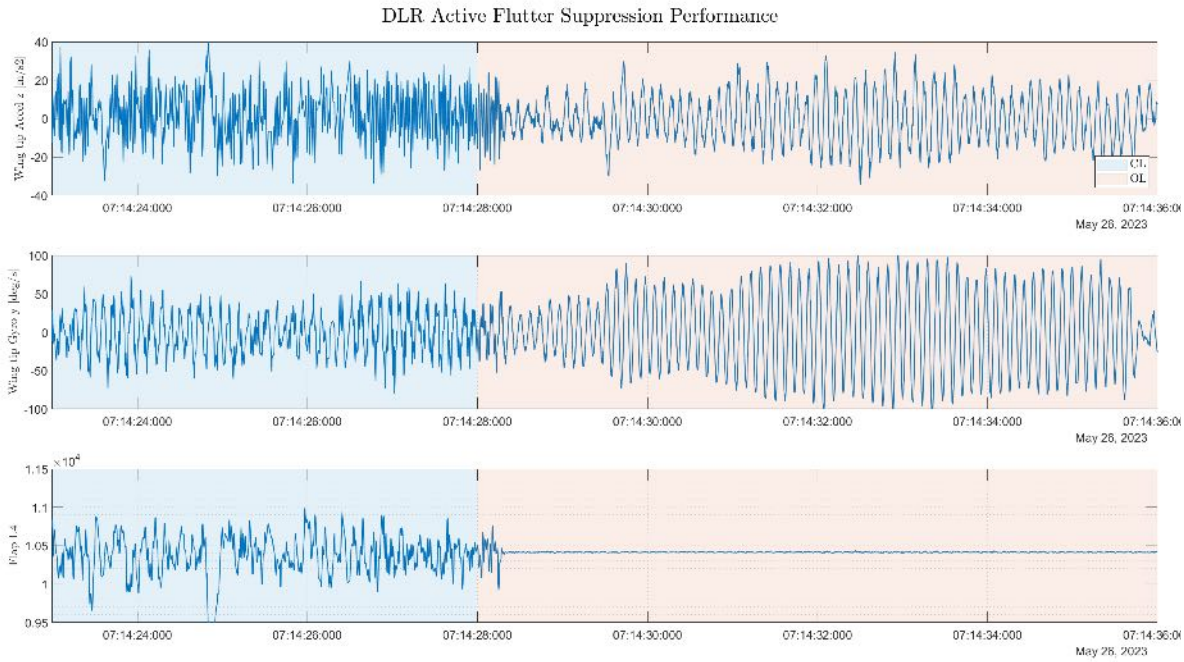


Figure 416: Flight Test Results(critical) - Time Series Plot for FT 35

.

fuselage sensor data along with rigid body actuator(Flap L3) is conducted as shown in figure 418. We can see that rigid body actuator i.e. Flap L3 is not active around higher flutter frequencies and similarly flutter actuator i.e. Flap L4(from figure 418) is not active at lower rigid body frequencies, ensuring minimal interaction of rigid and flutter controllers.

To summarize, the experimental validation of the DLR Active Flutter Suppression System through Flight Tests 33 and 35, demonstrated its effectiveness in suppressing flutter and expanding the flight envelope. The system showcased increased damping behavior and successfully mitigated flutter phenomena at subcritical and critical flutter velocities.

SZTAKI controller test results

In FT36, the aircraft was flown beyond the open-loop flutter speed, which is estimated to be $54\ \mathrm{m/s}$. Figures 419, 420, and 421 show the performance of the controller when the aircraft flies in the vicinity of $54\ \mathrm{m/s}$. The peak in acceleration in open loop at the flutter frequency is significantly damped in closed loop. This is especially noteworthy, since the controller only uses angular rate measurements.

Further results flying at $57\ \mathrm{m/s}$ and $59\ \mathrm{m/s}$ are shown in Figures 422, 423, and 424. In these cases, the signals captured in closed loop cannot be compared to open loop measurements since these are recorded beyond the flutter speed where safe operation in open loop in not possible. Oscillations in the vertical accelerations at the flutter frequency are still sufficiently suppressed. With the increasing speed, the high frequency vibrations become somewhat more sustained. Still, most of the control effort is focused on the neighbourhood of the flutter frequency.

There were two notable events during FT36. The first occurred, when a wind gust briefly but unexpectedly pushed the aircraft to $62\ \mathrm{m/s}$. This is significant because at such a high speed, the unwanted vibrations are bound to increase on the wings. Still, the controller maintained stability as shown in

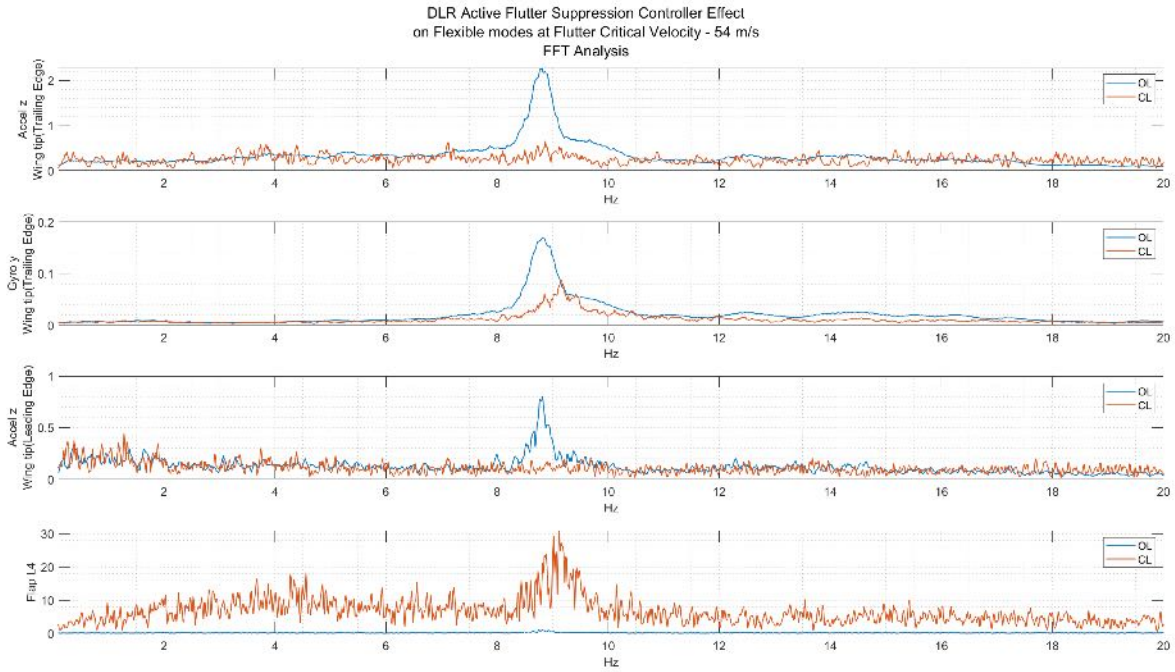


Figure 417: Flight Test Results(critical) - FFT Analysis

.

Figure 425. Sudden peaks in the accelerations and angular rates are observable at the time when the wind gust hit the aircraft but aside from that these values remain roughly between the same bounds as below $59\ \mathrm{m/s}$.

In the second event, the flutter controller is disengaged at $58\ \mathrm{m/s}$ and the flatter stopper is activated 2 seconds later to ensure that the potentially increasing oscillations are stopped before the airframe is damaged. This is illustrated in Figure 426. It has to be noted that by this point in this particular flight test, one of the flutter stoppers were damaged, which prevented the flutter from occurring. Therefore, the vibrations observable in the accelerations and angular rates are not actually due to flutter but are still structural oscillations. Nevertheless, the difference in the amplitude of the signals in Figure 426 is remarkable. After the controller is disengaged, the vibrations increase significantly, demonstrating that the controller indeed increased the damping while it was active.

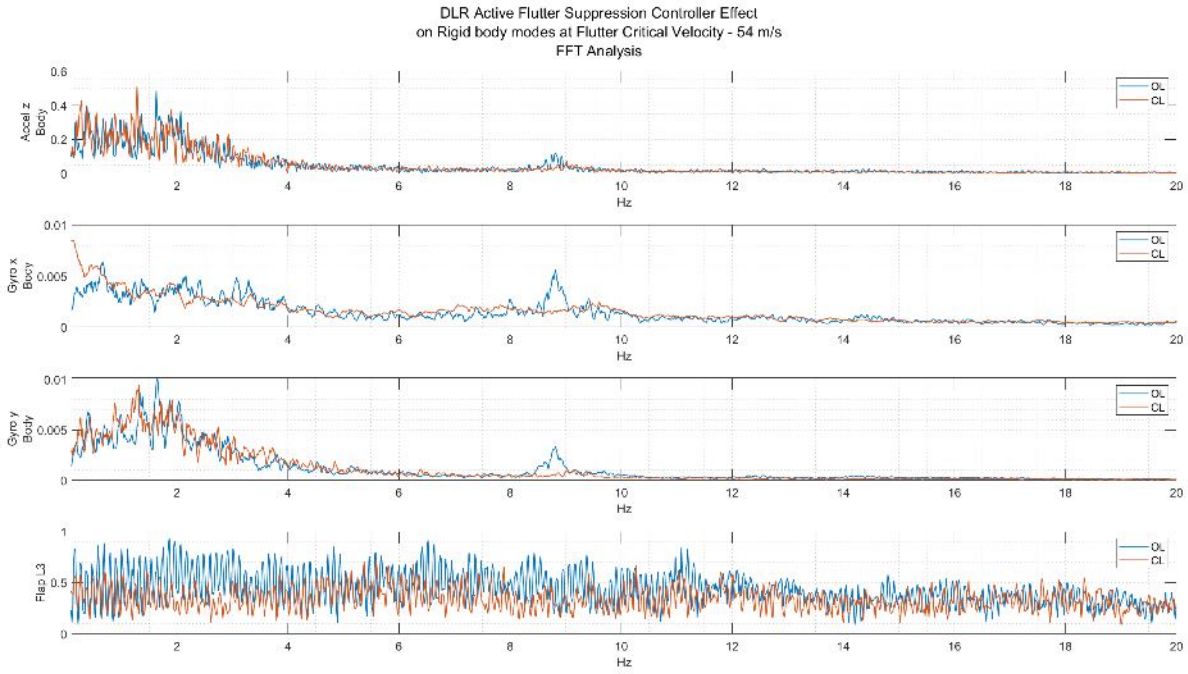


Figure 418: Flight Test Results(critical) - FFT Analysis

.

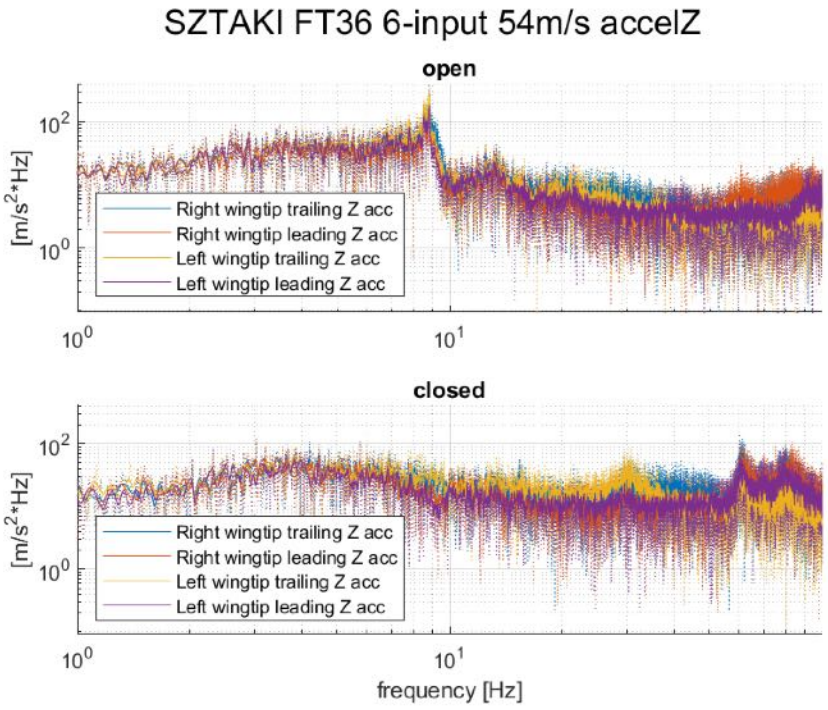


Figure 419: FT36 around 54 m/s: accelerations in the frequency domain.

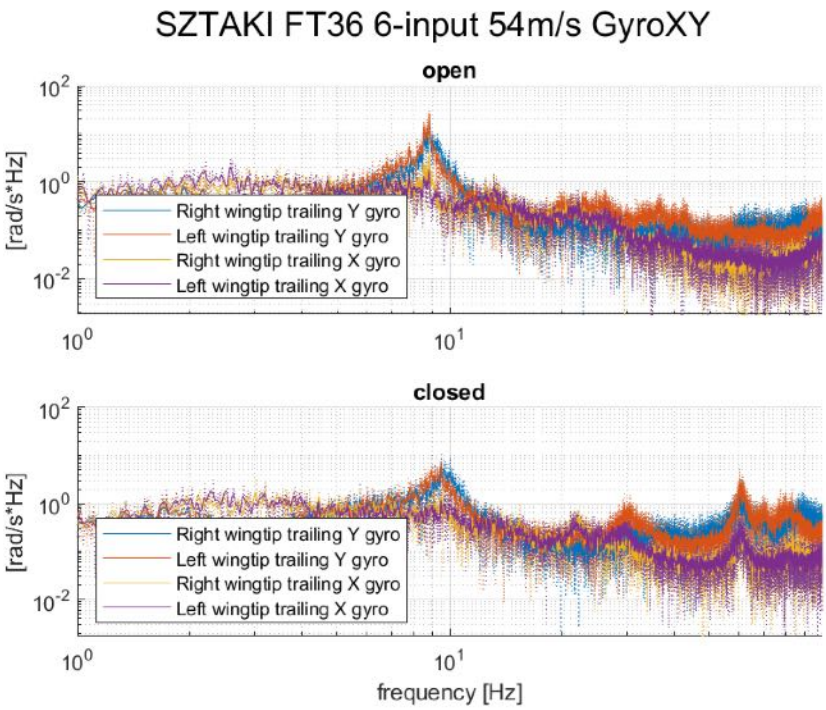


Figure 420: FT36 around 54 m/s: angular rates in the frequency domain.

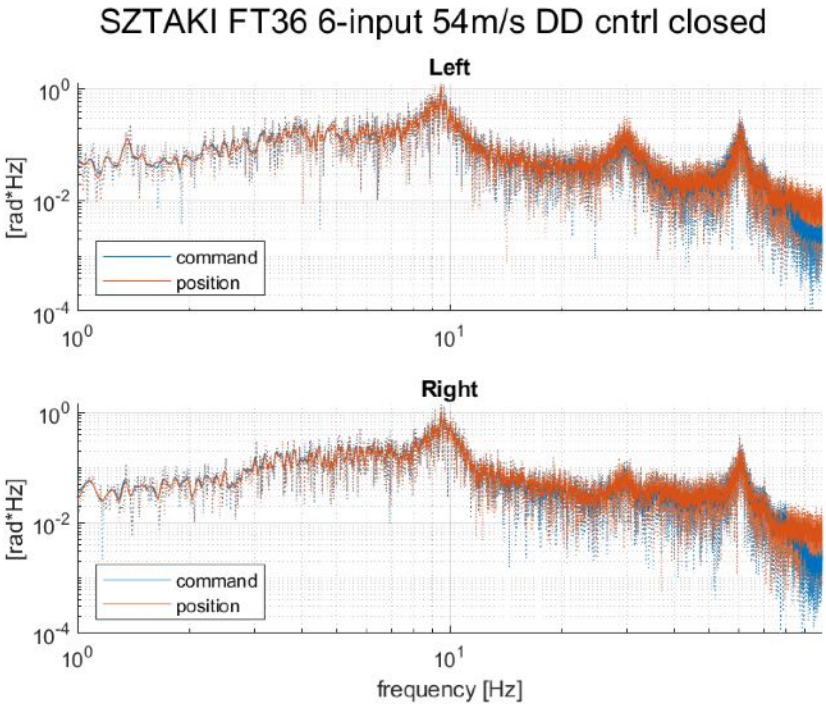


Figure 421: FT36 around 54 m/s: aileron command in the frequency domain.

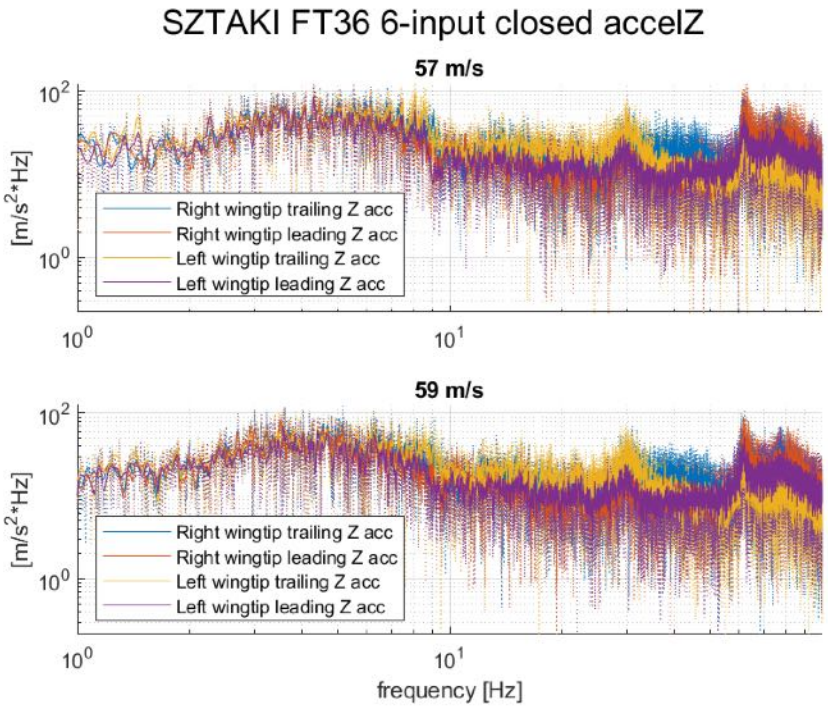


Figure 422: FT36 around 57 m/s and 59 m/s: accelerations in the frequency domain.

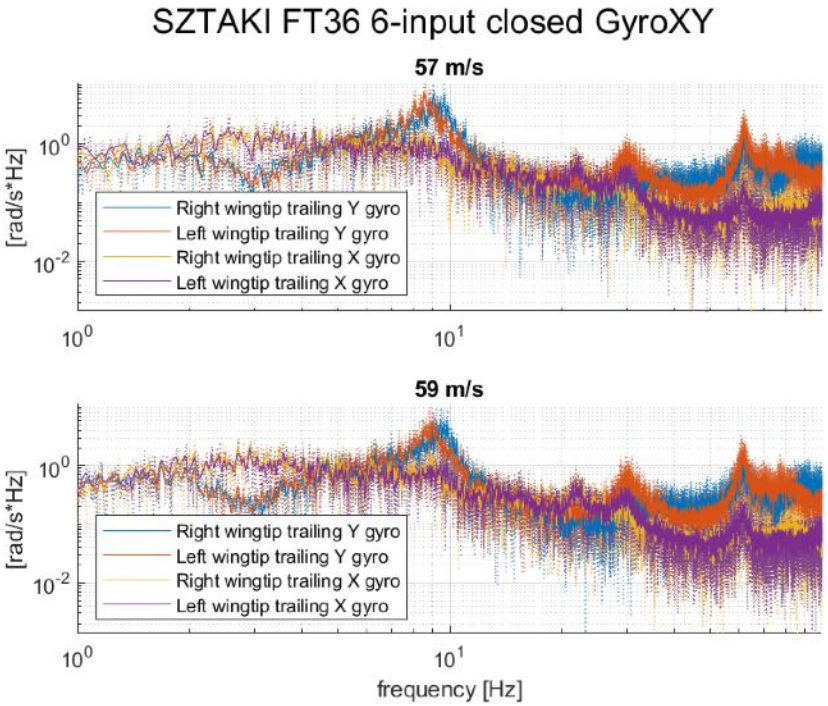


Figure 423: FT36 around 57 m/s and 59 m/s, angular rates in the frequency domain.

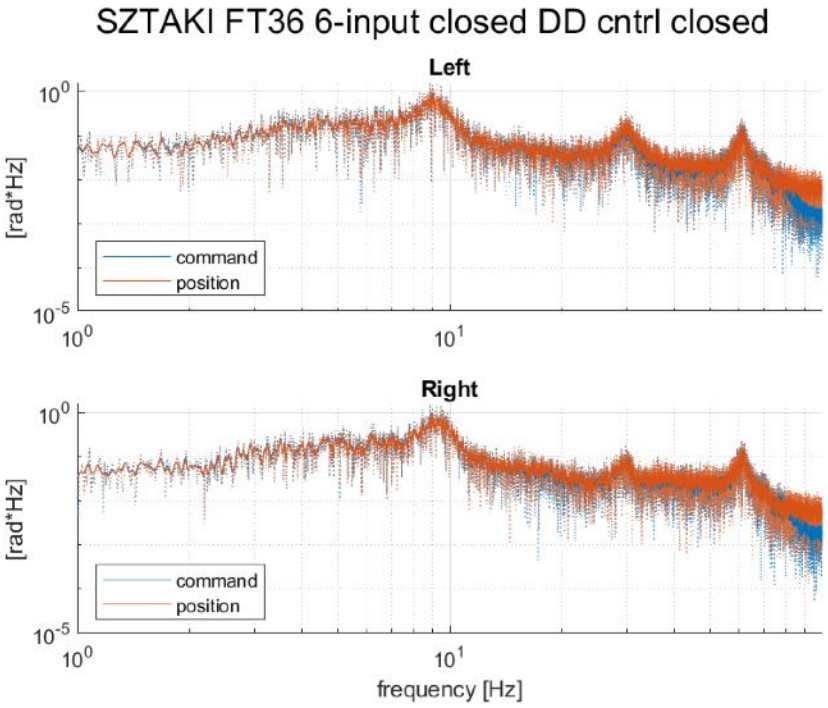


Figure 424: FT36 around 59 m/s, aileron command in the frequency domain.

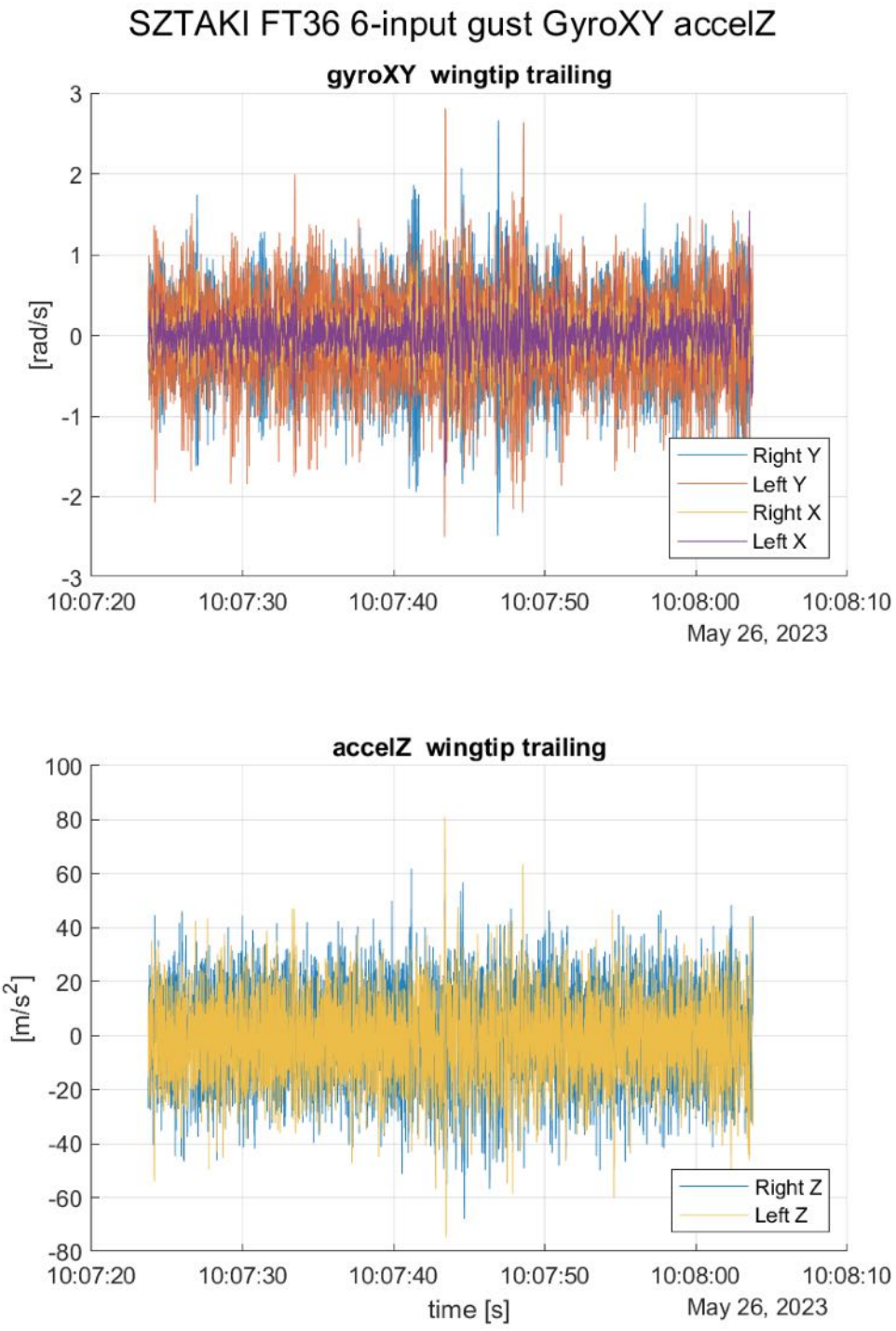


Figure 425: FT36 accelerations and angular rates during the wind gust.

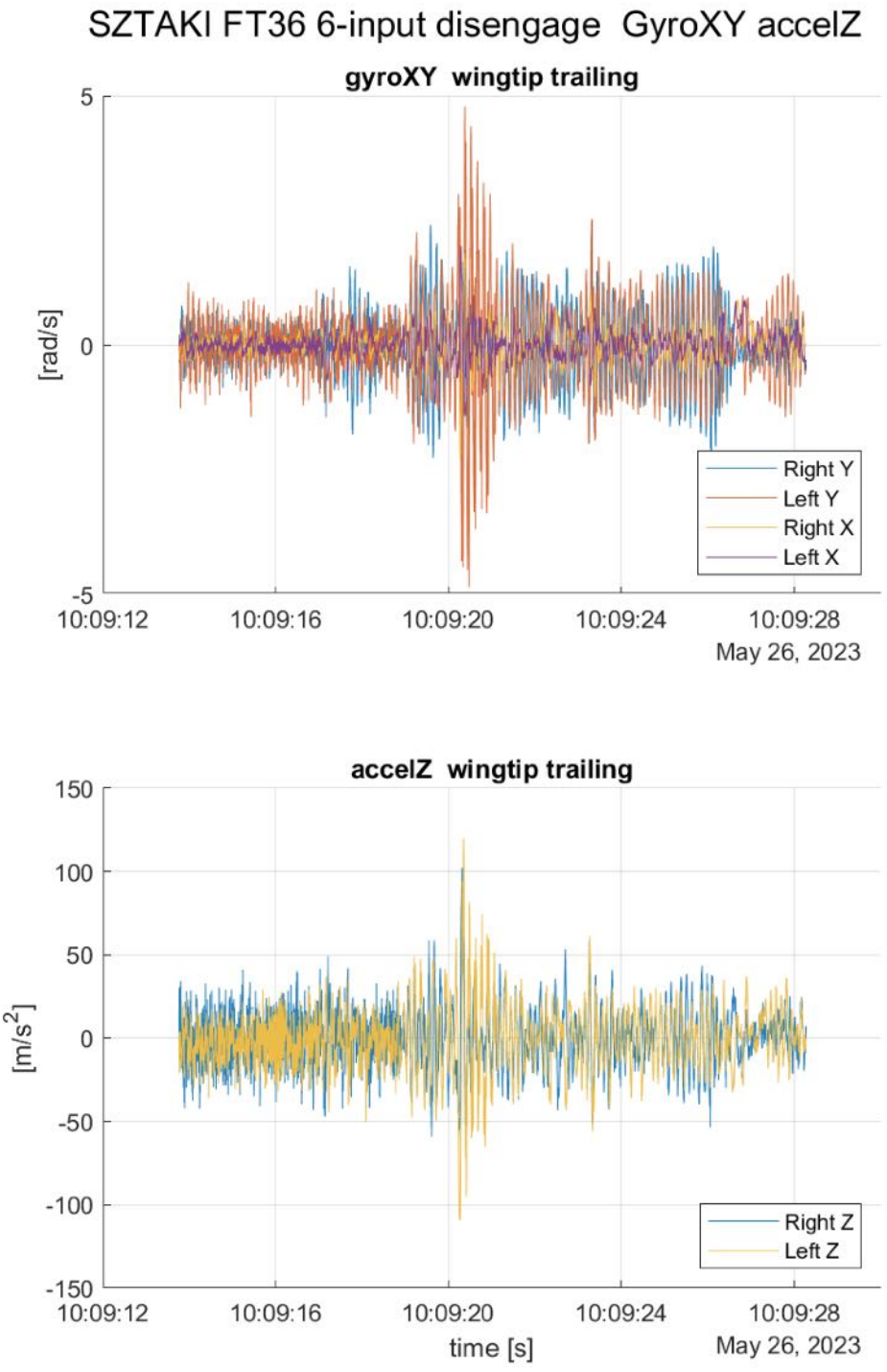


Figure 426: FT36 accelerations and angular rates during the controller disengagement.

Open-loop flutter test points (critical) (TUM, DLR, SZTAKI) After the successful deployment of the DLR and SZTAKI active flutter controllers beyond the predicted flutter boundary one important question remained:

1. are we sure that we have aeroelastic flutter at 56 m/s?

The OLM results indicate that we are very close to the flutter boundary at 54 m/s and that the trends of the flutter critical modes behave as expected. The updated simulations model agree with these trends and predict a flutter critical speed at 56 m/s. Nevertheless, the only way to confirm this with complete certainty is to fly open loop into the flutter boundary. Exactly this test was performed during FT37. The resulting time history from the flight can be seen in Figure 427. Two critical events were encountered at 407 s and 416 s. The result of the system identification with frequency and damping tracked during the flight can be seen in Figures 428 and 429. The results of reducing the time window buffer to 20 seconds and focusing on the stabilisation diagrams in the vicinity of the flutter critical events can be seen in Figure 430. The three modes of interest include the 2n wing bending in green, the 3n wing bending in purple and the symmetric wing torsion in yellow. Since the first flutter event occurs at 407 s the first buffer from 380 s to 400 s contains a stable system. In the second block from 385 s to 405 s we see that the symmetric torsion mode moves down in frequency and that the 2n wing bending moves up in frequency. The third block from 390 s to 410 s shows a further reduction in frequency of the symmetric torsion mode and a less reliable identification of the 2n wing bending mode. In the fourth block from 395 s to 415 s only 1 dominant mode at 8 Hz is identified. This phenomenon is also theoretically predicted and demonstrated in the paper by Özge Süelözgen [131].

Please note that these are preliminary results of work that is still in progress and will published together with the other project partners in the near future.

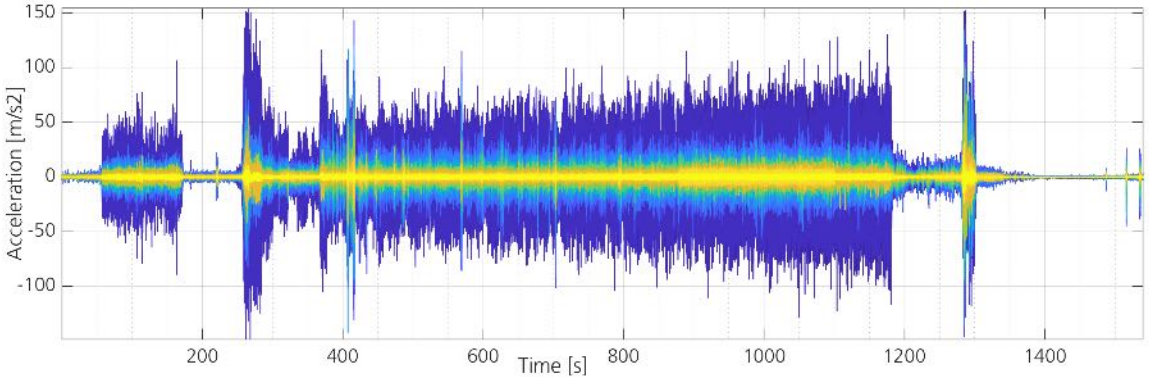


Figure 427: Acceleration time history from FT37.

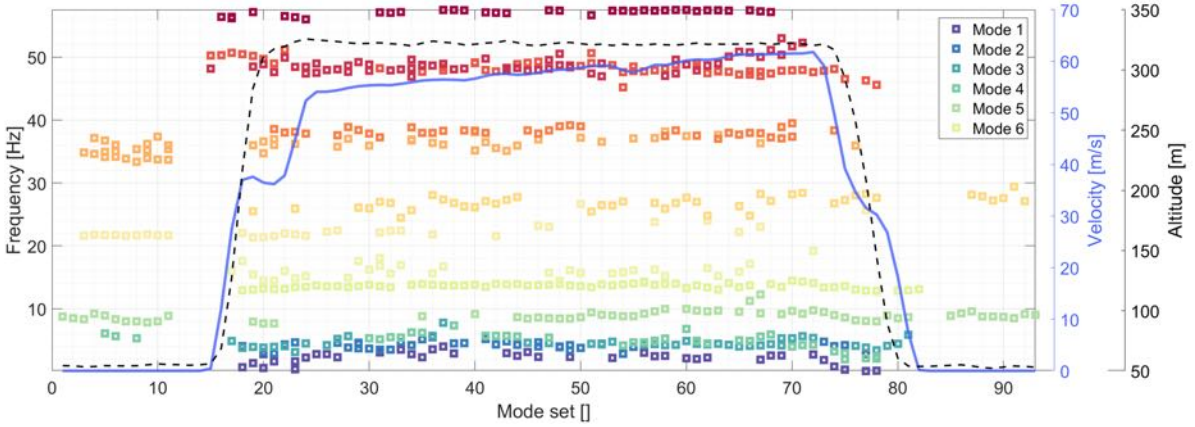


Figure 428: Eigenfrequency tracking from FT37.

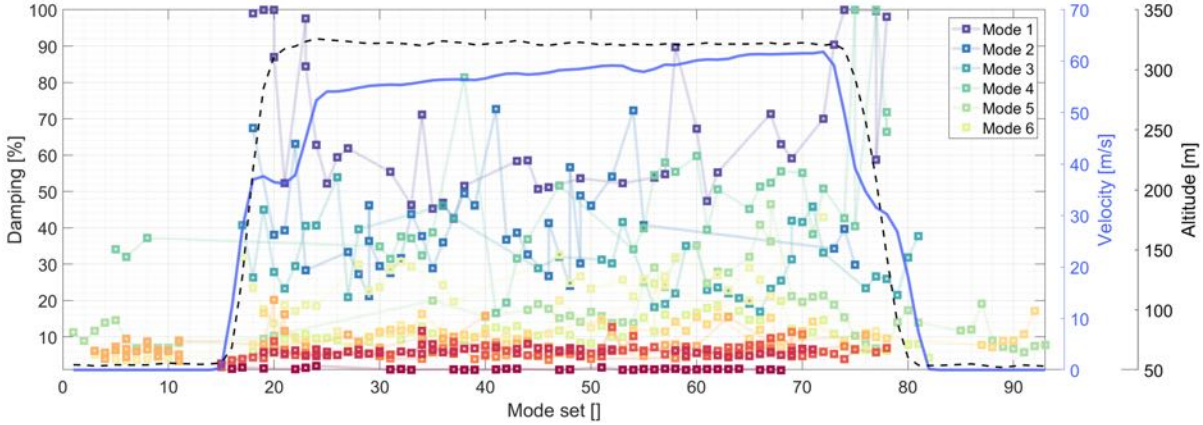


Figure 429: Damping tracking from FT37.

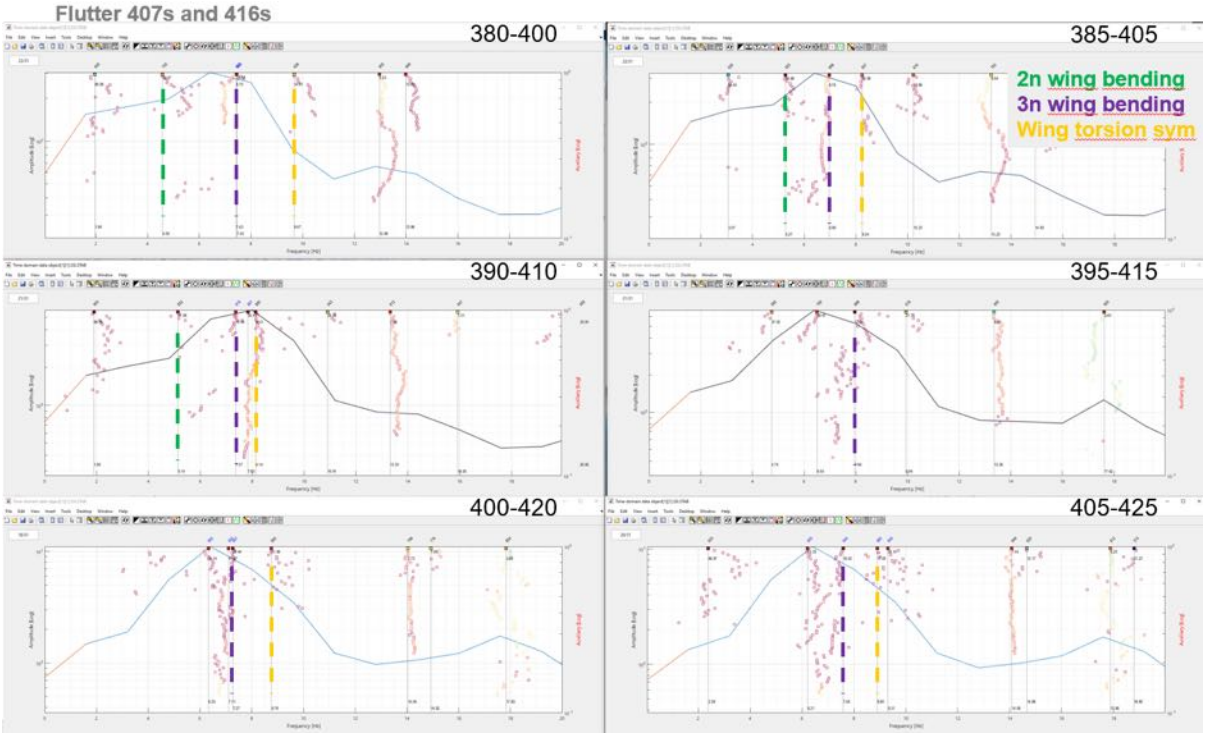


Figure 430: Stabilisation diagrams of 20 second data buffers during the flutter critical event.

Update of Linearised State-Space Models (DLR-SR)

1. Selected flight test data (manoeuvers, . . .)
2. Least-Square updating

Flight Testing of Flutter Controllers (DLR-SR, SZTAKI) Active Flutter Suppression control synthesis, verification and validation

In this section the Active flutter suppression controller synthesis and validation for the UAV is detailed. A nonlinear aeroelastic model of the demonstrator aircraft is derived, which serves as a basis for flutter suppression controller design and validation. For controller design, the H2-optimal blending approach is applied, which allows isolating the critical flutter modes even if they are within the same frequency range. As a result, each isolated flutter mode can be stabilized by a separate SISO controller which is scheduled with airspeed. A detailed description of the design, synthesis and tuning of the overall flutter suppression controller will be detailed. The promising results of the achieved flight envelope expansion, validated by flight tests, are discussed in following sections.

Controller Design and Synthesis

For the controller design, the nonlinear aeroelastic model of the UAV is linearized around steady horizontal flight at different airspeeds. Subsequently, a modal decomposition is carried out on each of the resulting LTI systems allowing for an individual analysis of the obtained aeroelastic modes. Both unstable modes describe flutter mechanisms based on a coupling of wing bending and torsion, where the mode shape is of symmetric and asymmetric nature, respectively. This is illustrated in a pole zero plot of the linearized aircraft model as shown in Figure 431. While the symmetric flutter mode becomes unstable at around 50 m/s with 8.0 Hz (50.3 rad/s), the asymmetric flutter mode follows at 52 m/s with 7.4 Hz (46 rad/s). To increase the aircraft's operational velocity range, it is hence required to stabilize the flutter modes, which is referred to as flutter suppression.

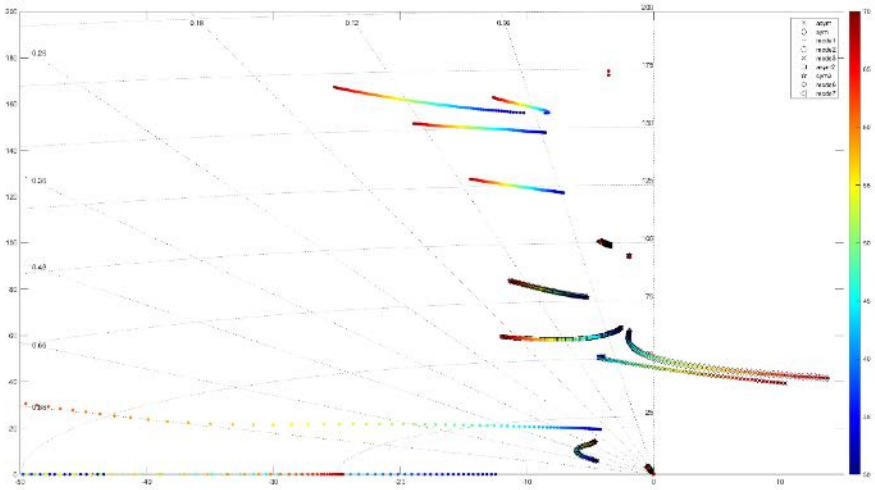


Figure 431: Pole migration

The two flutter modes, which are close in frequency but well distinguishable by their mode shapes, are decoupled and an individual mode stabilization by dedicated SISO controllers is enabled using

H2-optimal blending approach. The control inputs considered for active flutter suppression are the deflections commanded to the outermost aileron (powered by a high bandwidth direct drive actuation system) on each wing. As measurement signals, the pitch rate and vertical acceleration captured by the 13 different IMUs are considered. Since the mode shapes change only slightly within the critical airspeed range, it is sufficient to compute the blending vectors at a single airspeed of 60 m/s and hold them constant within the whole flight envelope. With the derived blending vectors, it is possible to design dedicated SISO controllers for the symmetric and asymmetric flutter mode. Furthermore. The SISO controller is augmented with a band-pass filter to ensure that no interference occurs with the flight control system operating at lower frequencies and that higher frequent modes are not excited. Since a large velocity range needs to be considered, the core of the flutter suppression controller is gain-scheduled with indicated airspeed. As explicit optimization constraints, a gain margin of 6 dB and a phase margin of 45◦ are demanded on the blended input channel at an indicated airspeed of 30, 40, 50 and 60 m/s. The resulting constrained multimodel optimization problems are non-convex and solved using Matlab's systune routine based on non-smooth optimization techniques.

Closing the two SISO loops (symmetric and asymmetric) stabilizes both flutter modes as it is illustrated in the following pole migration plot. The plot in Figure 432 compares the open-loop poles in gray to the closed-loop poles depicted in color dependent on the airspeed. Clearly visible is the unstable behavior, i.e., the crossing to the right half plane of the first (symmetric) and second (asymmetric) flutter mode in the open loop. With the flutter suppression controller, the symmetric flutter mode can be stabilized up to airspeeds of 62 m/s. The asymmetric mode is stabilized even beyond 70 m/s.

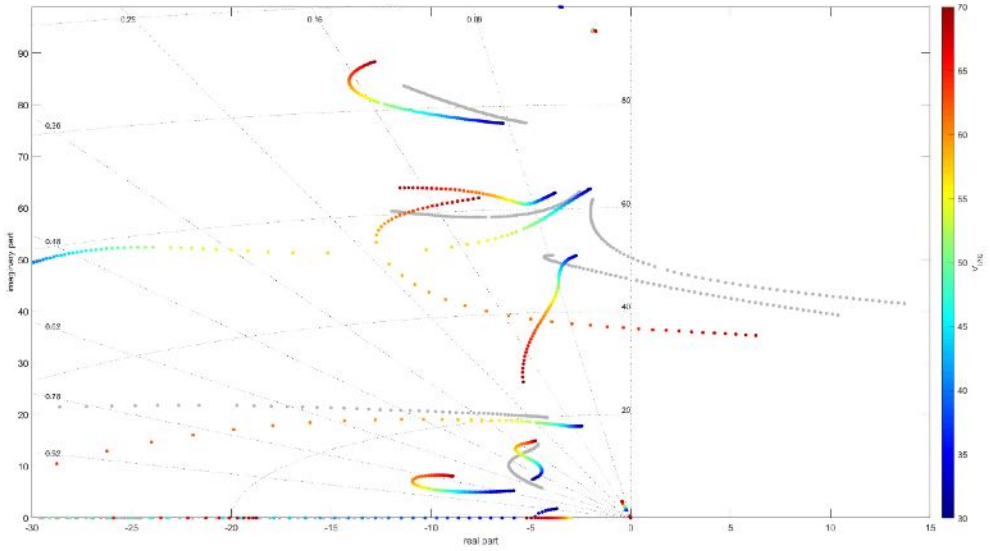


Figure 432: Pole migration of the closed loop

Flight test results

The following figures depict the flight test results.

At 46 m/s

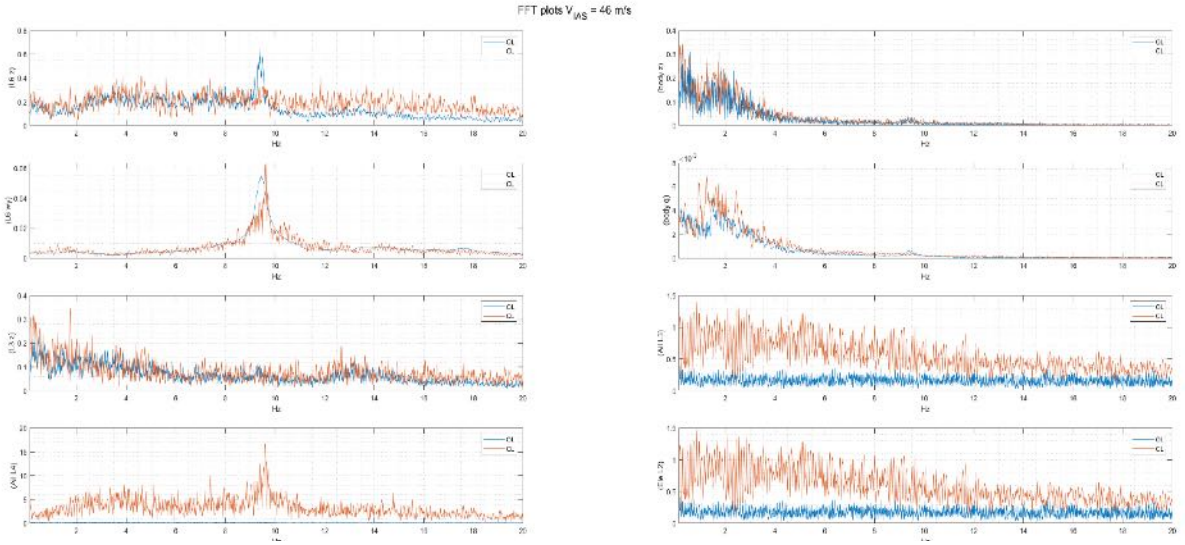


Figure 433: Flight test results at 46 m/s (a)

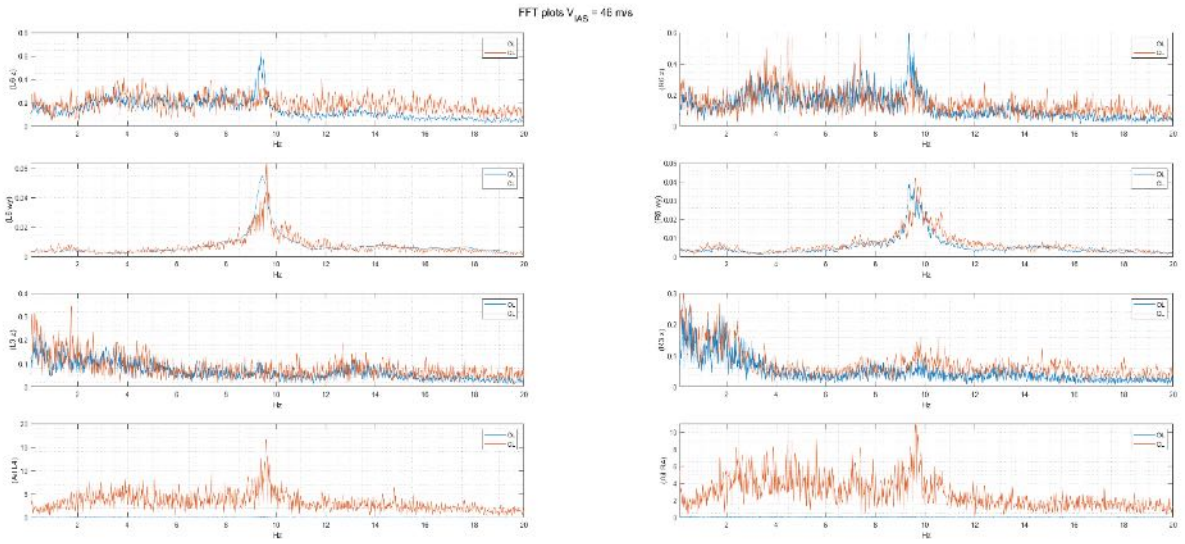


Figure 434: Flight test results at 46 m/s (b)

At 54m/s

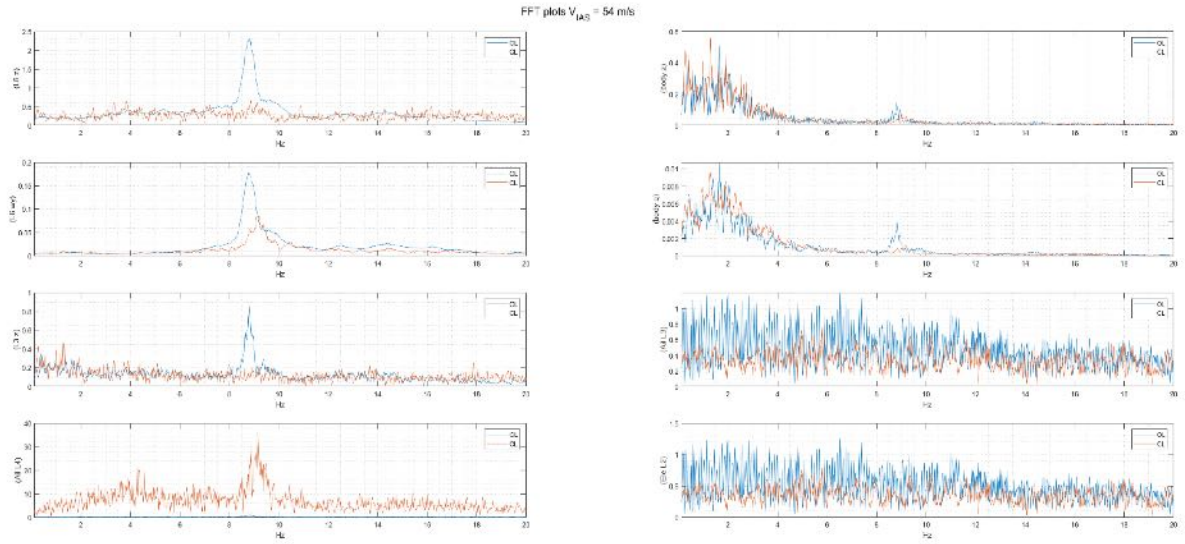


Figure 435: Flight test results at 54 m/s (a)

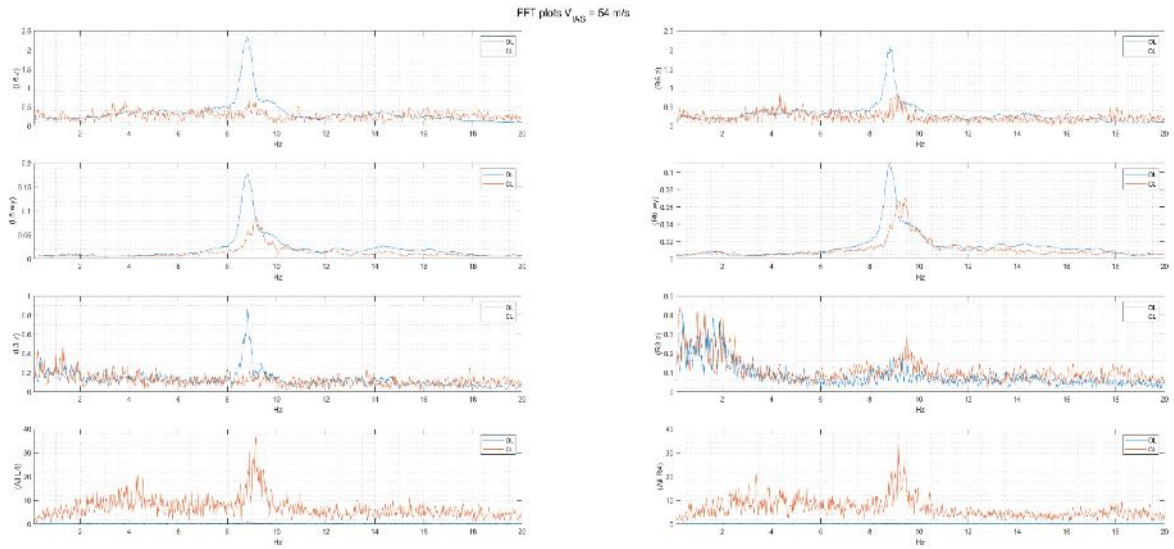


Figure 436: Flight test results at 54m/s (b)

Update and Validation of the -0 wing flight mechanical model Structural Dynamics

The structural dynamics of a flexible aircraft can be divided into rigid body and flexible body dynamics. The rigid body dynamics basically describe the manoeuver characteristics of the aircraft. In contrast, the flexible body dynamics represent the aircraft motion due to the flexibility of its structure. While the rigid body dynamics are described in nonlinear form, the equation of the flexible body dynamics is considered to be linear. A detailed FE model serves as basis for the structural model of the aircraft. The process of generating the FE model and its condensed version is described below. Subsequently, the EOM representing the rigid and flexible body dynamics are defined for the condensed model.

Finite Element Model

The aircraft structural FE model comprises the wing, fuselage and empennage and is shown in Figure 437. The FE software used here is MSC.NASTRAN. The wing is represented by a high-fidelity FE model comprising beam, surface and solid elements. Rigid body interpolation elements are added at predefined locations throughout the wing to facilitate the required model reduction. Further-more, the wing mass model is density-based as opposed to a lumped mass model. The fuselage structure is modelled using beam elements. The equivalent beam stiffnesses are obtained utilizing the cross sections of the fuselage hull at different sections and the lay-up of the hull. The mass is then lumped at the two beam nodes. The V-tail empennage FE model is shell-element-based comprising of the main structural load-bearing entities – the upper and lower skins, structural rubs, spars and the non-structural masses. Similar to the wing FE model, a density-based mass representation is used for the empennage as well.

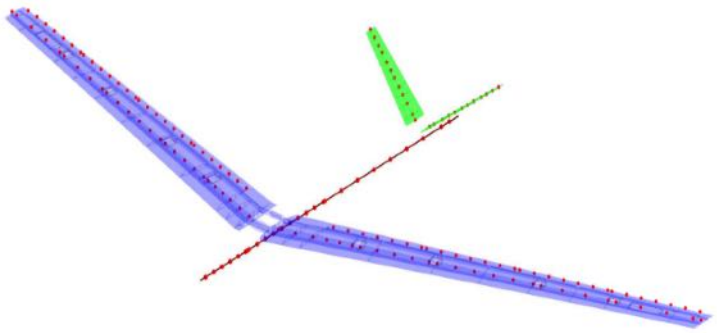


Figure 437: Full FE model of the FLEXOP demonstrator aircraft

Given that the FE model of the wing is of very high-fidelity (more than 600000 nodes), a Guyan reduction, also called condensation, is performed reducing the mass and stiffness matrix to less than 200 nodes in the condensed model.

Equations of Motion

The condensed model features rigid body and flexible modes, which are described by the EOM. These are based on an equilibrium of forces and moments. They describe the behaviour of the aircraft due to external loads originating from the aerodynamics and thrust. For simplification, the following assumptions are made.

1. As the earth rotation can be neglected, the inertial reference system is earth fixed.
2. Gravity is constant over the airframe.
3. The deformations of the airframe are considered to be small which allows the use of linear elastic theory defined by Hooke's law.
4. Due to small deformations of the aircraft structure, the aircraft mass moment of inertia J_b remains unchanged.

1. As the structural deformations are small, loads act on the undeformed airframe.
2. The eigenvectors of the modal analysis are orthogonal, because of which the total structural deformation can be written as a linear combination of the modal deflections.

3. The rigid body and flexible body EOM are considered to be decoupled.

Rigid Body Dynamics

For the derivation of the nonlinear flight mechanical EOM, the aircraft is considered as a rigid body with a constant mass m_b and constant mass moment of inertia J_b. Therefore, the aircraft rigid body motion is described by the Newton-Euler EOM

$$\begin{bmatrix} m_b(\dot{V}_b + \Omega_b \times V_b - T_{be} g_e) \\ J_b \dot{\Omega}_b + \Omega_b \times (J_b \Omega_b) \end{bmatrix} = \underbrace{\Phi_{gb}^T P_g^{\text{ext}}(t)}_{P_b^{\text{ext}}(t)} = \begin{bmatrix} F \\ M \end{bmatrix}, \quad (1)$$

In Equation (1) the translational and angular velocity of the aircraft with respect to the body frame of reference are given by V_b and Ω_b. The vector g_e represents the gravitational acceleration, which is transformed with T_{be} from the earth-fixed to the body-fixed frame of reference. The external loads vector

$$P_g^{\text{ext}}(t) = P_g^{\text{eng}}(t) + P_g^{\text{aero}}(t) \quad (2)$$

includes the loads acting on the aircraft structure. Here the loads due to the engine thrust $P_g{}^{eng}$(t) and the aerodynamic loads $P_g{}^{aero}$(t) are considered. By means of the matrix Φ_{gb}^T the external loads are transformed into the rigid body frame.

Flexible Body Dynamics

As the displacements due to the aircraft flexibility are assumed to be small, linear elastic theory is applied to define the flexible body motion. Therefore the correlation between external loads $P_g{}^{ext}$(t) and the generalized coordinates u_f representing the modal deformation of the structure is given by the differential equation

$$M_{ff}\ddot{u}_f + B_{ff}\dot{u}_f + K_{ff}u_f = \underbrace{\Phi_{gf}^T P_g^{\text{ext}}(t)}_{P_f^{\text{ext}}(t)}. \quad (3)$$

The matrices M_{ff}, B_{ff} and K_{ff} depict the modal masses, dampings and stiffnesses. The modal matrix Φ_{gf} contains the eigenvectors of the structural modes sorted by frequency. Typically, higher frequencies have a smaller contribution to the overall system performance. Consequently, modal truncation can be applied to reduce the DOF significantly by considering only the most relevant eigenmodes.

Aerodynamics

The aerodynamic loads represent the major external loads acting on the aircraft structure. Their calculation is based on the VLM for steady aerodynamics and the DLM for unsteady aerodynamics. Both methods are based on a panel model, which is described in the following section.

Panel Model

The lifting surfaces are discretised by several trapezoidal-shaped panels, known as aerodynamic boxes as shown in Figure 438. Of note is the panel model for the fuselage. The wetted areas of the fuselage

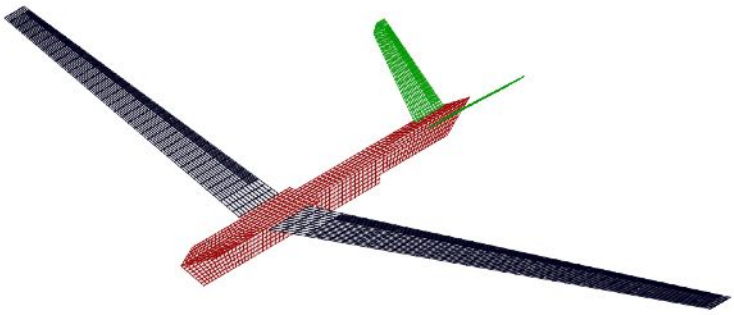


Figure 438: Aerodynamic boxes of the FLEXOP demonstrator aircraft

are projected onto a T-cruciform shaped panel model. Although this is a vast simplification, the fuselage aerodynamics are modelled quite accurately with respect to higher-order CFD simulations.

Steady Aerodynamics

The VLM is used to model steady aerodynamics. As can be seen in Figure 439a, each aerodynamic box of the panel model possesses a horseshoe vortex at point l on the quarter-chord line. Due to the Helmholtz theorem the vortex is shed downstream to infinity at the side edges of the box. For each aerodynamic box the Pistolesi Theorem needs to be met, stating that there is no perpendicular flow through the control point j at the three-quarter-chord line.

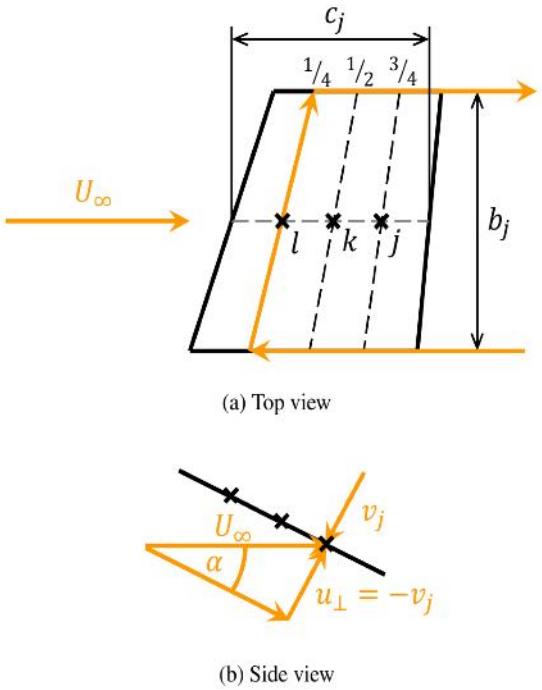


Figure 439: Schematic drawing on an aerodynamcs box

Therefore the induced velocity at the control point needs to equalize the perpendicular component of the incoming flow, like shown in Figure 439b. By means of the Biot-Savart law the induced velocities v_j

due to the circulation strengths Γ_j of the horseshoe vortices can be determined by

$$v_j = A_{jj}\Gamma_j. \tag{4}$$

The matrix A_{jj} describes the contribution of all vortices to the induced velocities of the aerodynamic boxes. Inverting A_{jj} and multiplying with $2 = c_j$, where c_j is the chord length of the respective aerodynamic box, leads to the aerodynamic influence coefficient (AIC) matrix Q_{jj}. In the steady aerody-namic case it is considered constant. The pressure coefficient c_{pj} of a panel is then determined by

$$\Delta c_{pj} = Q_{jj} w_j, \tag{5}$$

where w_j is the velocity normalized with the flight speed U_∞. It is assumed to be equal to the angle of attack α_j of a panel, i.e. $w_j = \sin(w_j)$, as only small angles are considered. The downwash w_j comprises different aerodynamic contributions. It is affected by a rigid body motion of the aircraft with

$$w_{jb_1} = \frac{c_r}{2U_\infty} D_{jk,2} \Phi_{ka} T_{ab} \begin{bmatrix} V_b \\ \Omega_b \end{bmatrix}. \tag{6}$$

The vector $[V_b^T \quad \Omega_b^T]^T$ contains the rigid body velocities V_b and angular rates Ω_b and is transformed to the aerodynamic centre by means of T_{ab}. Subsequently, the respective motion of each panel reference point k is calculated by multiplying Φ_{ka}. The resulting contribution to the downwash is then determined by multiplication with the matrix $D_{jk;2}$ and factorisation with $c_r/2U_\infty$, where c_r depicts the reference chord length. Further details on the determination of the contributions of the downwash can be found in Ref. 10. Under the assumption of small angles Equation (6) can be rearranged to

$$\begin{aligned} w_{jb_1} &= \frac{c_r}{2} D_{jk,2} \Phi_{ka} \begin{bmatrix} \cos\alpha_a \cos\beta_a \\ \tan\beta_a \\ \sin\alpha_a \\ p_a/U_\infty \\ q_a/U_\infty \\ r_a/U_\infty \end{bmatrix} \\ &\approx \frac{c_r}{2} D_{jk,2} \Phi_{ka} \begin{bmatrix} 1 \\ \beta_a \\ \alpha_a \\ p_a/U_\infty \\ q_a/U_\infty \\ r_a/U_\infty \end{bmatrix}. \end{aligned} \tag{7}$$

It can be seen that, besides the angular rates p_a, q_a and r_a, the downwash is affected by the sideslip angle β_a and the angle of attack α_a. The "1" in the vector represents a constant contribution to the downwash. This gives the opportunity to add the downwash caused by effects like camber and twist by adaptation of the first column of Φ_{ka}. As a first step, it is updated based on a steady computational fluid dynamics (CFD) calculation. The deflection of the control surfaces u_x is taken into account by changes in the downwash

$$w_{jx_0} = D_{jk,1}\Phi_{kx}u_x. \tag{8}$$

The matrix Φ_{ka} links control surface deflections to the corresponding aerodynamic boxes. The differentiation matrix $D_{jk,1}$ then relates a displacement of the panel reference point k to the downwash w_j. Besides, the control surface deflection rate alters the lift, which can be accounted for by

$$w_{jx_1} = \frac{c_r}{2U_\infty} D_{jk,2}\Phi_{kx}\dot{u}_x. \tag{9}$$

As depicted in Equation (3) the structural dynamics are affected by the aerodynamic loads ${P_g}^{aero}$. These can be expressed in terms of w_j as

$$P_g^{\text{aero}} = q_\infty T_{kg}^T S_{kj} Q_{jj} w_j + q_\infty T_{ag}^T S_r c_D, \tag{19}$$

where the second term represents the aerodynamic drag loads with reference area S_r and the transformation matrix from the mean aerodynamic centre to the structural grid T_{ag}^T. Matrix S_{kj} depicts an integration relating the pressure in the aerodynamic boxes at point j with the forces at the aerodynamic grid points k. The forces at the aerodynamic grid points k are then interpolated onto the structural grid points via the transpose of the spline matrix T_{kg}. The splining model of the wing is exemplary shown in Figure 440.

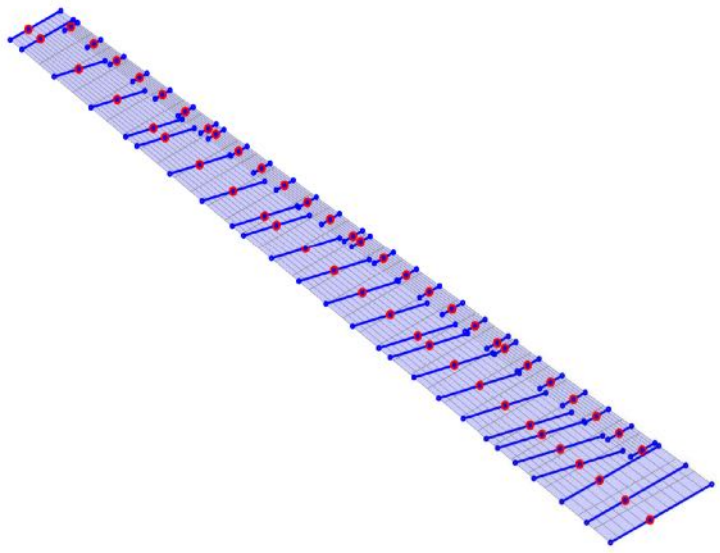


Figure 440: Splining between the aerodynamic model and structural model of the right wing

Multiplying with the dynamic pressure q_∞ then leads to the aerodynamic loads acting on the structure. To distinguish between the distributions of the aerodynamic loads to the rigid and flexible body dynamics, Equation (19) is multiplied by Φ_{gb} and Φ_{gf} leading to

$$\begin{bmatrix} P_b^{\text{aero}} \\ P_f^{\text{aero}} \end{bmatrix} = \begin{bmatrix} \Phi_{gb}^T \\ \Phi_{gf}^T \end{bmatrix} P_g^{\text{aero}}. \tag{20}$$

Due to aerodynamic loads the aircraft structure performs rigid body and flexible body motions which, in turn, affect the aircraft aerodynamics. Therefore the aeroelastic model is considered a loop between structural dynamics and aerodynamics.

Performed Manoeuvers

The parameter estimation process strongly depends on the performed manoeuvers in flight, as they define how well the characteristics of the aircraft can be determined. In order to define suitable excitation signals, a priori knowledge on the model is used. However, this is conflicting as the accuracy of the examined model determines the quality of the model parameters to be estimated. Nevertheless, under the assumption, that the chosen modelling process provides realistic results, this approach is considered applicable.

Flight Mechanical Manoeuvers

The first goal is to update the flight mechanical model. Therefore all contributions resulting from the aircraft flexibility are neglected. Besides, unsteady aerodynamic effects are ignored, as their contribution to the flight mechanical model is assumed to be small. As a result, for the aerodynamic load P_b^{aero} on the right-hand side of the rigid body equation of motion (1) it is only accounted for the downwash w_{jb1}, w_{jx0} and w_{jx1} yielding to

$$\begin{aligned} P_b^{\text{aero}} =& q_\infty \underbrace{\Phi_{gb}^T T_{kg}^T S_{kj} Q_{jj} \frac{c_r}{2} D_{jk,2} \Phi_{ka}}_{DQ_{h,b1}} \begin{bmatrix} 1 \\ \beta_a \\ \alpha_a \\ p_a/U_\infty \\ q_a/U_\infty \\ r_a/U_\infty \end{bmatrix} \\ &+ q_\infty \underbrace{\Phi_{gb}^T T_{kg}^T S_{kj} Q_{jj} D_{jk,1} \Phi_{kx}}_{DQ_{h,x0}} u_x \\ &+ q_\infty \underbrace{\Phi_{gb}^T T_{kg}^T S_{kj} Q_{jj} \frac{c_r}{2U_\infty} D_{jk,2} \Phi_{kx}}_{DQ_{h,x1}} \dot{u}_x. \end{aligned} \tag{21}$$

It is assumed, that the correlation between the control surfaces and the aerodynamic load given by the matrices DQ_{h,x_0} and DQ_{h,x_1} is accurately predicted by the proposed model. The focus is on an update of the rigid body contribution gathered in matrix DQ_{h,b_1}. It is a 6x6-matrix with the entries

$$Dh_{b1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & f_{y\beta} & 0 & f_{yp} & 0 & f_{yr} \\ f_{z0} & 0 & f_{z\alpha} & 0 & f_{zq} & 0 \\ 0 & m_{x\beta} & 0 & m_{xp} & 0 & m_{xr} \\ m_{y0} & 0 & m_{y\alpha} & 0 & m_{yq} & 0 \\ 0 & m_{z\beta} & 0 & m_{zp} & 0 & m_{zr} \end{bmatrix}. \quad (22)$$

There are 15 parameters, that are to be estimated. As can be seen the parameters related to forces in x-direction are neglected, due to the mentioned constraints of the model. Many more entries are equal to zero or considered too small to have a significant influence on the parameter estimation. The remaining parameters can be associated with either a longitudinal or a lateral aircraft motion.

In Table 2 the performed manoeuvers, separated in longitudinal and lateral, are listed with the parameters, that mainly contribute to the aircraft motion.

Table 2: Performed manoeuvres for the parameter estimation [11]

Longitudinal	
Steady level flight:	$f_{z0}, f_{z\alpha}, m_{y0}, m_{y\alpha}$
Pushover-pullup:	$f_{z0}, f_{z\alpha}, f_{zq}$
Short period:	$f_{z\alpha}, f_{zq}, m_{y\alpha}, m_{yq}$
Phugoid:	$f_{z0}, f_{z\alpha}, f_{zq}, m_{y0}, m_{y\alpha}, m_{yq}$
Lateral	
Steady sideslip:	$f_{y\beta}, m_{z\beta}$
Dutch-roll:	$f_{y\beta}, f_{yr}, m_{x\beta}, m_{xr}, m_{z\beta}, m_{zr}$
Bank-to-bank:	f_{yp}, m_{xp}, m_{zp}

The definition of the excitation signals for the short period, phugoid and dutch-roll mode are determined based on an a priori analysis of the initial model. The phugoid is excited by an elevator pulse, that is chosen to last 2 seconds with an amplitude of approximately 3°. This elevator deflection was found to be appropriate to excite the phugoid mode of the aircraft. The dutch-roll mode can be excited by a doublet on the rudder. The amplitude is chosen to be around 3°, while the half time length $\Delta t_{doublet}$ of the doublet is calculated with the dutch-roll frequency $\omega_{dutch-roll}$ by the rule of thumb

$$\Delta t_{\text{doublet}} \approx \frac{2.3}{\omega_{\text{dutch-roll}}} \quad (23)$$

to be 1.22 seconds. The dutch-roll frequency $\omega_{dutch-roll}$ is determined from the simulation in advance of the flight test. Equivalently, the short period mode can be observed by exciting the elevator with a doublet. Equation (23) gives a $\Delta t_{doublet}$ of 0.24 seconds with the pre-determined frequency of the short-period $\omega_{short-period}$. The amplitude is chosen to be around 6°.

The steady level flight, pushover-pullup, steady sideslip and bank-to-bank manoeuvers were flown manually by the pilot.

Update of the Rigid Body Model

When it comes to updating an aircraft model or rather specific model parameters, a suitable process needs to be set up. On the one hand a model structure must be given including parameters to be estimated and on the other hand an optimization algorithm to find the somewhat best model parameters needs to be given. There exist different optimization algorithms to estimate model parameters, like the output error method (OEM), the filter error method (FEM) and more. Within the scope of this paper the output error method based on maximum likelihood estimation is chosen.

Output Error Method

In Figure 441 the basic procedure of the OEM is shown.

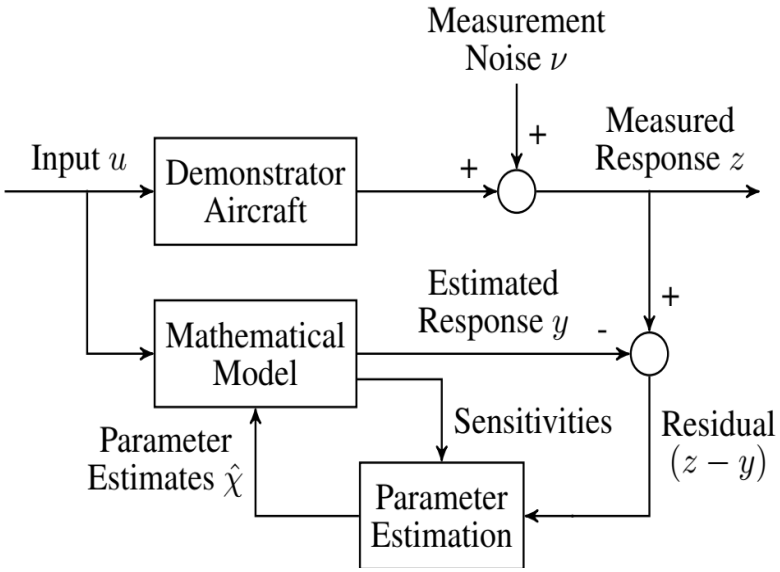


Figure 441: Concept of the output error method

The upper path represents the flight test, where the outcome is the measured inputs and outputs. The OEM assumes, that the outputs are affected by measurement noise. Process noise, however, is neglected. Subsequently, the inputs are fed into the mathematical model to conduct a simulation of the considered flight test manoeuver. Based on the difference between the flight test measurements and the simulation outputs, the parameters of the mathematical model are updated by means of an optimisation.

It is assumed, that the model equations are given in the form of

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), \chi),\ x(t_0) = x_0 \\ y(t) &= g(x(t), u(t), \chi) \\ z(t_k) &= y(t_k) + \nu(t_k). \end{aligned} \tag{24}$$

The first two equations describe the proposed mathematical model. They dependent on the desired parameters χ. The last equation provides the relation between the discrete flight test measurements z and the output of the measurement equation y at a time instant t_k. They exclusively differ in the

measurement noise ν. The noise process is considered stochastic and is characterized by Gaussian white noise with zero mean. Its definition is

$$\begin{aligned} E\{\nu(t_k)\} &= 0 \\ E\{\nu(t_k)\nu(t_l)^T\} &= R\delta_{kl}. \end{aligned} \tag{25}$$

The second expression of Equation (25) suggests that the noise process is white noise, as it is time independent. Simultaneously the amplitude depends on chance defined by a Gaussian distribution with covariance matrix R it describes Gaussian noise. As a result the measurement vector $z(t_k)$ with dimension n_z is affected by Gaussian white noise and therefore its values are assumed to be Gaussian distributed with a probability density function

$$\begin{aligned} p(z(t_k)|\chi) =& \frac{1}{(2\pi)^(n_z/2)sqrt|R|} \\ & \exp\left(-\frac{1}{2}(z(t_k)-y(t_k))^T R^{-1}(z(t_k)-y(t_k))\right). \end{aligned} \tag{26}$$

With respect to Equation (24) the expected value of z(t_k) is assumed to be $E\{z(t_k)\} = y(t_k)$ for the model parameters χ. For a set of N measurements the likelihood function becomes

$$\begin{aligned} p(z(t_1),...,z(t_N)|\chi) &= \prod_{k=1}^{N} p(z(t_k)|\chi) \\ &= [(2\pi)^{n_z}|R|]^{-N/2} \\ \exp\left(-\frac{1}{2}\sum_{k=1}^{N}(z(t_k)-y(t_k))^T R^{-1}(z(t_k)-y(t_k))\right). \end{aligned} \tag{27}$$

Goal of the maximum likelihood method (MLM) is to identify the model parameters χ, which maximise the probability defined by Equation (27). The optimal solution is the maximum likelihood estimate obtained as

$$\begin{aligned} \hat{\chi}_{ML} &= \arg\left\{\max_{\chi} p(z|\chi)\right\} \\ &= \arg\left\{\min_{\chi}\left(-\ln p(z|\chi)\right)\right\}. \end{aligned} \tag{28}$$

For greater ease of handling the negative logarithm of the likelihood function $p(x|\chi)$ is considered, which simplifies Equation (27) to the cost function

$$J(\chi, R) = \frac{1}{2}\sum_{k=1}^{N}\left((z(t_k) - y(t_k))^T R^{-1}(z(t_k) - y(t_k))\right) + \frac{N}{2}\ln(|R|) + \frac{Nn_z}{2}\ln(2\pi). \tag{29}$$

At this point it is assumed, that the covariance matrix R is unknown a priori. As R depends on the model parameters χ and vice versa, the relaxation strategy is used to find the optimal solution of the redefined likelihood function (29) in two steps. Firstly, for a given parameter vector χ the maximum likelihood estimate of R is obtained by setting the partial derivative $\partial J(\chi, R)/\partial R$ to zero. This yields

$$R = \frac{1}{N}\sum_{k=1}^{N}(z(t_k) - y(t_k))(z(t_k) - y(t_k))^T. \tag{30}$$

Secondly, substituting (30) in (29) provides

$$J(\chi) = \frac{1}{2}n_z N + \frac{N}{2}\ln(|R|) + \frac{Nn_z}{2}\ln(2\pi). \tag{31}$$

Apart from ln(|R|) all terms in Equation (31) are independent from the model parameters χ. The cost function therefore reduces to

$$J(\chi) = \det(R). \tag{32}$$

Equation (32) is solved iteratively for the optimal model parameter χ by means of a Gauss-Newton algorithm.

Two-Step Method

By means of the two-step method (TSM) the model parameters can be determined. The TSM divides the state and parameter estimation problem in a flight path reconstruction and a parameter identification part. The flight path reconstruction is used to accurately reconstruct the time history of the aircraft states during the manoeuver and besides allows the determination of potential instrumentation errors. As some sensor readings, like the angle of attack and the airspeed, are prone to be inaccurate, the measurements are improved based on past, present and future data and the flight mechanical equations. Subsequently, the identification of the model parameters follows.

The success of the TSM depends on the aircraft to be tested, the aircraft instrumentation, the excitation signals, the mathematical model selected for identification and the chosen algorithm for the analysis and adaption of the model.

Flight Path Reconstruction

The flight path reconstruction is based on a non-linear state-space system consisting of flight mechanical state and measurement equations. The considered inputs are the translational accelerations a_{bm} and the rotational rates Ω_{bm} measured in flight by an IMU placed in the fuselage. The states are the velocity vector V_b, the Euler angles ϕ, θ, ψ and the altitude h. The resulting state equations are given by

$$\dot{V}_b = a_b - (\Omega_{b,m} - \Delta\Omega_b) \times V_b + T_{be} g_e \tag{33}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\sin\theta} & \dfrac{\cos\phi}{\sin\theta} \end{bmatrix} (\Omega_{b,m} - \Delta\Omega_b) \tag{34}$$

$$\dot{h} = [0 \quad 0 \quad -1]\, T_{be}^{-1} V_b. \tag{35}$$

Starting point of the state equations is the equilibrium of forces of the rigid-body equation of motion. Solving Equation (1) for $\dot{V}_b$ leads to Equation (33), where Ω_b is replaced by its flight test measurement Ω_{bm} including a potential sensor bias $\Delta\Omega_b$. The translational acceleration a_b is given with respect to the center of gravity. It is determined by

$$a_b = a_{b,m} - \dot{\Omega}_b \times d_s - \Omega_b \times (\Omega_b \times d_s) - \Delta a_b. \tag{36}$$

The acceleration measurement $a_{b,m}$ needs to be corrected for the coriolis and the centrifugal force caused by the offset between the acceleration sensor position and the center of gravity d_s. A potential sensor bias is covered by Δa_b. Additional state equations of the Euler angles ϕ, θ, ψ are considered through Equation (34). The remaining state equation is given by Equation (35). The inverse of T_{be} transforms the velocity V_b to the Earth-fixed frame of reference. Extracting only the element, which contributes to the z-direction, and changing the sign leads to the derivative of the altitude $\dot{h}$.

The outputs or reconstructed instrumentation measurements are the true airspeed $U_{\infty,r}$, the angle of attack α_r, the sideslip angle β_r, the Euler angles ϕ_r, θ_r, ψ_r and the altitude h_r. The corresponding measurement equations are given by

$$U_{\infty,r} = \|V_b\|_2 \tag{37}$$

$$\alpha_r = K_\alpha \tan^{-1}\left(\frac{V_{nb,z}}{V_{nb,x}}\right) + \Delta\alpha \tag{38}$$

$$\beta_r = K_\beta \sin^{-1}\left(\frac{V_{nb,y}}{|V_{nb}|}\right) + \Delta\beta \tag{39}$$

$$\phi_r = \phi \tag{40}$$

$$\theta_r = \theta \tag{41}$$

$$\psi_r = \psi \tag{42}$$

$$h_r = h. \tag{43}$$

As the α and β measurements of the noseboom are sensitive to errors, the scaling and bias variables K_α, $\Delta\alpha$, K_β and $\Delta\beta$ are introduced. The velocity vector V_{nb} at the noseboom is determined by

$$V_{nb} = V_b + (\Omega_{b,m} - \Delta\Omega_b) \times d_{nb}, \tag{44}$$

where d_{nb} is the distance between the aircraft CG and the noseboom. In theory the difference between the flight test measurements and the reconstructed measurements in (37-43) with respect to the OEM is only coming from the process noise ν.

The unknown parameters $\Delta\Omega_b$, Δa_b, K_α, $\Delta\alpha$, K_β and $\Delta\beta$ as well as the initial states V_{b0}, $[\phi_0\, \theta_0,\, \psi_0]^T$, h_0 in Equations (33-35) are determined based on the introduced OEM algorithm. The residual (z-y) to be minimized is the difference between the flight test measurements and their reconstructed counterpart in Equations (37-43).

The FPR is performed for each considered manoeuver type separately. Figures 442 and 443 depict the FPR exemplary for a pushover-pullup manoeuver (POPU) and for a sideslip manoeuver (SL) in comparison with the measured flight test data (FTD).

Only the measurement variables that play a major role for the manoeuvers are shown. For the POPU manoeuver it can be seen, that α changes dynamically, while the remaining measurements are rather smooth. Nevertheless, the reconstructed α follows very closely the measurement.

The sideslip manoeuver is not performed at a constant β as intended. However, it still offers the opportunity for updating lateral model parameters. The FPR follows the trends of the observations very well. An exception is the reconstructed true airspeed U_∞ which follows the trend of the measurement, but does not change as dynamically. As this behaviour is not observed for the additional measurements, it is valid to say the true airspeed is more strongly affected by disturbance.

Parameter Estimation

The parameter estimation is the second step of the two step method. The control surface deflections commanded during the various flight test manoeuvers are fed in the rigid body equation of motion (1). As mentioned before the parameters of the matrix $DQ_{h,b1}$ defined in Equation (21) are to be estimated. Based on the comparison between the outputs of the flight test z and the simulation y the model parameters are updated like described in the section ”Output Error Method”.

The parameters corresponding to the longitudinal and lateral motion respectively are updated in separate steps. At first, the lateral manoeuvers are used to improve the matrix $DQ_{h,b1}$ with respect to the

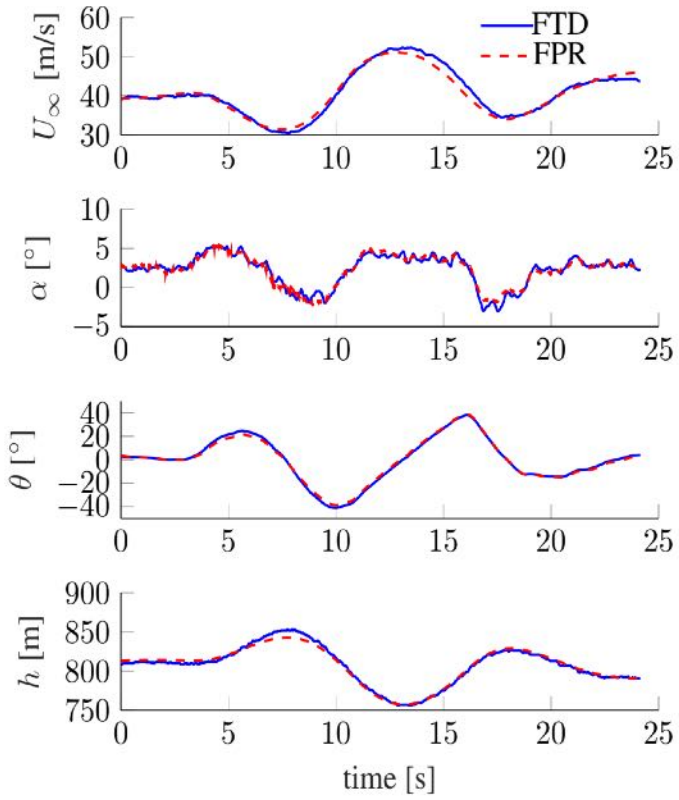


Figure 442: Comparison of reconstructed and flight test measurements (POPU)

parameters $f_{y,\beta}$, $f_{y,p}$, $f_{y,r}$, $m_{x,\beta}$, $m_{x,p}$, $m_{x,r}$, $m_{z,\beta}$, $m_{z,p}$, $m_{z,r}$. Subsequently, the longitudinal parameters $f_{z,0}$, $f_{z,\alpha}$, $f_{z,q}$, $m_{y,0}$, $m_{y,\alpha}$ and $m_{y,q}$ are updated with the matrix $DQ_{h,b1}$ coming from the previous step. The final step is to redo the lateral update. This approach is chosen, because the longitudinal manoeuvers also feature lateral contributions and vice versa. Therefore, a strict separation of the manoeuvers is not possible.

At the end, the OEM leads to the parameters summarized in Tables (3) and (4).

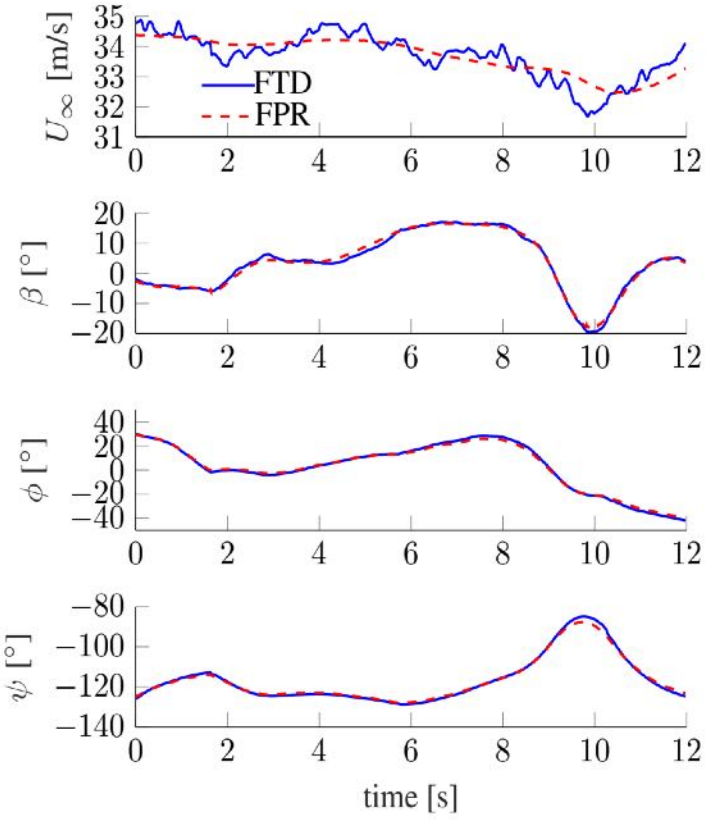


Figure 443: Comparison of reconstructed and flight test measurements (SL)

Table 3: Initial and final longitudinal parameters

Parameter	Initial	Final
f_{z0}:	-0.601	-0.484
$f_{z\alpha}$:	-15.12	-15.19
f_{zq}:	-7.41	-0.056
m_{y0}:	0.067	-0.02
$m_{y\alpha}$:	-1.63	-2.19
m_{yq}:	-13.58	-5.94

Table 4: Initial and final lateral parameters

Parameter	Initial	Final
$f_{y\beta}$:	-0.621	-0.661
f_{yp}:	-0.321	-2.35
f_{yr}:	2.52	3.41
$m_{x\beta}$:	-0.382	-2.08
m_{xp}:	-109.9	-126.4
m_{xr}:	1.28	32.61
$m_{z\beta}$:	0.464	0.48
m_{zp}:	0.294	-3.95
m_{zr}:	-4.02	-4.05

Of note is that the $f_{z,0}$ contributing to the lift with respect to camber and drag was lightly overestimated with the CFD calculations mentioned before. For corresponding moment coefficient $m_{y,0}$, however, undergoes a relatively big change and switches sign. The $f_{z,\alpha}$ and $f_{y,\beta}$ value does not change much, which proves the strength of the VLM/DLM modelling approach. Some final parameter values differ strongly from their initial values. It is still under investigation to what extent the simplified modelling of the x-forces plays a role.

When the pushover-pullup (POPU) manoeuver is performed with the model featuring the estimated parameters (PE), one can recognize a strong similarity with the reconstructed flight test data (FPR). Figure 444 depicts the trend of some of the observation variables affected by a longitudinal motion.

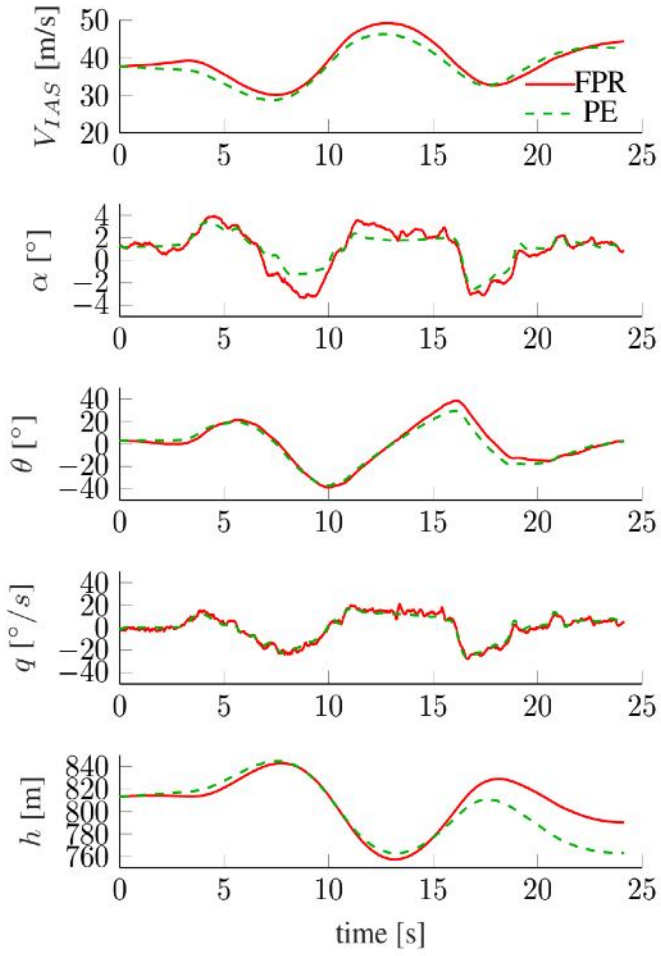


Figure 444: Comparison between reconstructed and simulated measurements (POPU)

The difference between the reconstructed and simulated angle of attack α possibly reveals the sensitivity to external disturbances. However, especially the pitch rate q matches very well between both data sets.

The measurements of the sideslip manoeuver exhibited in Figure 445 proves, that the set of estimated parameters of the model fits well with the flight test data.

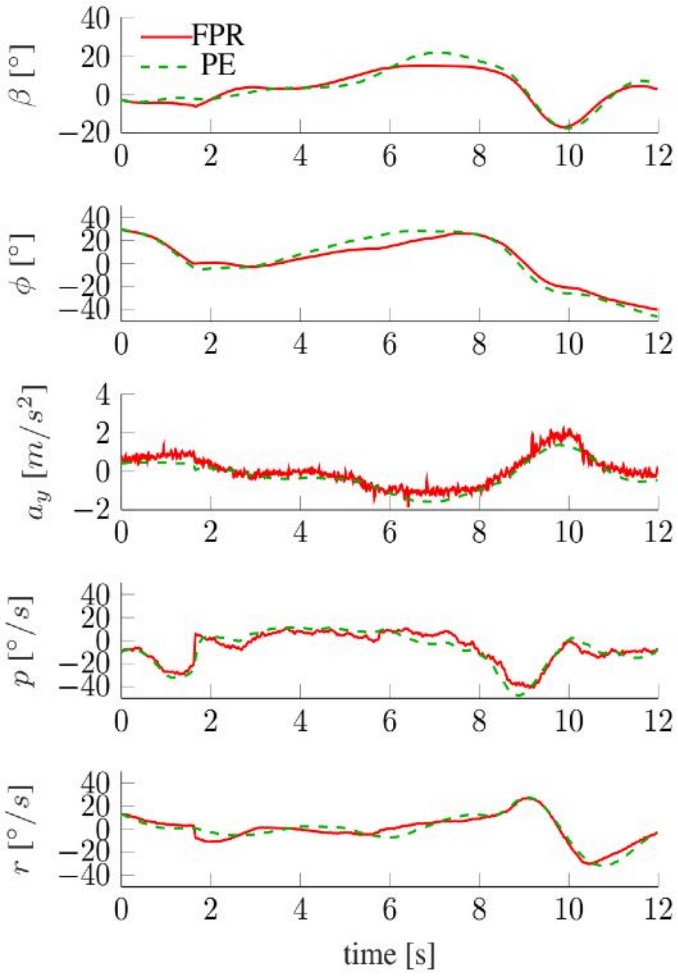


Figure 445: Comparison between reconstructed and simulated measurements (SL)

2.3.4 Deviations, their reason, impact on the project and corrective actions

Many unforeseen problems were realized in 2020 and 2021. These were: landing gear, engine and ground control issues, and findings that the aerodynamics of the actual aircraft are not as predicted. Additionally, to these technical problems there were severe Corona restrictions prohibiting to work in the workshop or conducting flight tests for a longer period of time.

None of these challenges were easy to cope with, therefore only two flights were made in phase 1. The biggest problem came from not being able to fix the aircraft's ground controllability in a timely manner. This mainly was due to two reasons: there were restrictions imposed on access to the workshops at TUM and the Airport due to Corona, and the problem appeared to be way more difficult to solve than was initially anticipated. The many iterations, implementation of which only started in June, could only improve bit by bit. In addition, not having a workshop at the airport, this proved time-costly to try out new concepts.

Due to the landing gear problems, it was decided not to risk the aircraft and not attempt to conduct test flights as was planned before. Therefore, the first flight test campaign had to be postponed until the controllability of the aircraft is sufficient, and it can be made sure that the aircraft will not be destroyed due to ground controllability problems (Figure 446).

Figure 446: Another close call due to the inadequate steering on the ground. The demonstrator stopped shortly before the taxiway lamp

Flight test data and further aerodynamic analysis revealed that the modelled aircraft generates significantly less lift than initially predicted. Revised flap settings for take-off and landing configurations

significantly improved the take-off and landing performance and solved this problem and adding also up to a better ground handling end especially much shorter takeoff run.

In January 2021 the rules for flight permits for UAVs in Germany have changed. The organization which handles the permits changed from the Bavarian to the National Aviation Authority of Germany (LBA). This increases the processing times of applications. Moreover, and more importantly, the UAV flight safety rules which were applied to T-FLEX at the beginning of FLEXOP project in 2018 have now changed significantly. The flight permit is only issued if the risk assessment process (SORA), provided by the EASA (`https://www.easa.europa.eu/document-library/easy-access-rules/online-publications/easy-access-rules-unmanned-aircraft-systems?page=4%23%5FToc18667479`), is followed. These rules not only are more detailed than before, but they are also stricter.

It was decided to perform a flight test campaign at Cochstedt UAV Flight Test Center (which belongs to the DLR) end of September or beginning of October 2021. The application process for a flight permit was started mid-August. At that moment, the LBA announced that it could take around 6 weeks to process it. It was soon realized the big amount of work needed to submit the application. The new rules meant that flight areas had to be recalculated, further simulations (for the parachute) be done, manuals updated. Risk analysis had to be rewritten. The final application (100 pages long) was finally submitted on 30.09.2021, after having multiple feedback talks with the LBA. The first reply with request for further corrections of application was received on 26.10.2021. It meant that the flight campaign for this year had to be cancelled. After more iterations, we finally managed to get the flight permit for Cochstedt shortly before the second flight campaign in 2022.

However, as this airport is not in near vicinity of Munich, where the flight crew is based, organization of flight tests did become more difficult. It was not possible to get a permit for Oberpfaffenhofen within the scope of this project, as the conditions with more populated area are even more difficult.

The next topic to address is the -3-wing design. It was late due to delays in the development of the multidisciplinary design loop. Therefore, a study was conducted that investigated the feasibility to equip an existing -0- or -2-wing with the hardware necessary to serve as a -3-wing substitute. The feasibility investigation focused on the feasibility of attaching and actuating a total of eight flaps at each wing, as that number of flaps was considered sufficient to allow shape control and load alleviation, while still feasible to integrate into an existing wing.Based on aforementioned considerations, the retrofit of either a -0- or -2-wing is considered feasible and was implemented.

Due to an accident with the demonstrator in August 2022 during the second flight test campaign, the fuselage had to be rebuilt. This accident also led to a major reorganization of the project flight test objectives and the consortium made a common decision to prioritize the wing-1 flight tests leading to active flutter control experiments, after the demonstrator is rebuilt, rather than proceeding with the wing-3 tests. Hence, the advanced wing manufacturing and integration, aimed at demonstrating active wing shape control via movable trailing edge flaps, was paused, and the major focus shifted to fuselage and demonstrator rebuild.

To avoid similar events and to improve ground handling and improve ease of operational experience, several design changes were made to the original system. Compared to the first iteration of the fuselage, all the design change made on subsystem level and cabling layout proved to be a positive step forward to an easily usable demonstrator. The introduction of programmed interface for the pilots, and the different switches on the payload area, allowed a faster and more stable turn-on and turn-off process. The fuel tank extension made it possible to make 30-40 minutes flights, thus allowing to make the most out of each flight. Although the rebuilding of the aircraft took a lot of time and effort from the staff, the improvements to the system led to the successful completion of all planned active flutter control experiments of the -1 campaigns. In overall, the P-FLEX called new aircraft proved to be a reliable platform that now can be operated efficiently and safely.

2.4 Explanation of the work carried out per WP - Work Package 4

2.4.1 Objectives and activities

Within FliPASED the focus lies on including control design as a primary discipline in a collaborative design workflow/MDA/MDO.

Some previous experience within DLR included already a comprehensive load process (Klimmek, et al., 2017) Digital-X and Victoria (Görtz, et al., 2016; Görtz, et al., 2020) and preliminary steps have been taken to consider active control systems within the design cycle (Ilic, et al., 2020).

The efforts within the Filpased project seek to focus on inclusion of the control part, while deemphasizing the aerodynamic part.

In FliPASED the aerodynamics will consist mainly of low fidelity aerodynamics and methods based on potential flow theory. Hence, transsonic effects like shocks and wave drag will not be considered in the scale up task. This is a conscious choice in order to avoid overlap with other projects and to allow quick calculation times. Furthermore, the choice of an MDO architecture plays a secondary role.

This distinguishes the approach in FliPASED to other efforts which mainly focus on aero-structural optimization (Kenway, et al., 2014). In the future the findings of FliPASED may be integrated in MDO workflows with more realistic aerodynamic properties.

FliPASED wants to demonstrate the benefits of including active control technologies early in the design rather than an afterthought.

Scale up objective function

The overall objective function for the scale up task will be based on evaluation of mission criteria, such as range or blockfuel. This way two primary design goals can be addressed.

The first goal is to minimize the aerodynamic drag. Specifically, the induced drag is addressed by high aspect ratio wing designs. However, the resulting slender wing structures tend to be very flexible and defueling the wing tanks change the mass distribution and in turn the shape of the wing. To counteract the detrimental effect on the induced aerodynamic drag, active wing shape control deflects the control surfaces to restore a drag optimal lift distribution for the changing wing mass.

The second goal is to minimize the structural weight. This can be achieved by employing active load alleviation control laws to minimize design loads for manoeuvres as well as gusts and turbulence in combination with passive methods for load alleviation such as aeroelastic tailoring.

Furthermore, the aforementioned high aspect ratio wings are more prone to an adverse fluid structure interaction called flutter. Conventionally, this is addressed by increasing the wing stiffness or placing additional mass in suitable locations. The employment of active flutter suppression allows to relax these stiffness requirements and therefore save weight.

To assess the benefits of the mentioned active control technologies, the mission is analyzed at multiple points of the mission, i.e. different mass cases due to defueling. The conjecture is that inclusion of active control theory in the design phase leads to very different wing designs and large overall fuel savings.

2.4.2 Starting point and approach

The workflow that is setup in WP2, initially addresses the design of wings for the demonstrator. The objective there is to maximize the difference between open loop and closed loop performance of the individual control functions in order to assess and validate their benefits by flight test. Fuel burn and minimal weight are not primary design objectives.

For the scale up task, a passenger aircraft is considered. The design objectives have been described in the previous section. Apart from the differing objective functions, the most notable difference to the demonstrator workflow, is that the structure is now sized by the loads, i.e. the employed control functions have a direct impact on the overall weight of the structure. The changed stiffness and mass properties therefore make a convergence loop necessary.

Figure 447 shows an early version of the envisaged scale up workflow. The XDSM diagram shows a convergence loop including structural sizing, controller design of the various functions and the loads analysis of the closed loop aircraft.

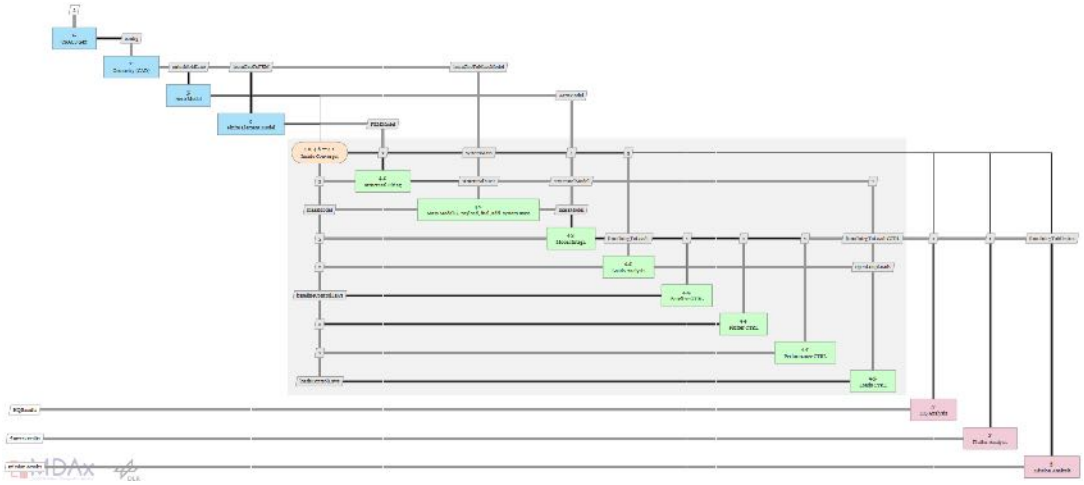


Figure 447: XDSM diagram of the scale up workflow

A further complication arises, as the CATIA based structural model generation is targeted towards the demonstrator wing. It will be investigated how this model generation process can be adapted to a transport aircraft wing. As contingency, an alternative model generation module (CPACS-MONA) is available at DLR's Institute of Aeroelasticity. This module has been used in several MDO workflows before.

Figure 448: Airbus XRF1 FEM model

2.4.3 Efforts and achieved results, name of involved partners

So far different reference aircraft models have been evaluated based on their suitability for the scale up task.

XRF1: Airbus eXternal Research Forum Model (A330 like)

The XRF1 Model is a multidisciplinary aircraft model which is intended to further development and validation of flight physics and broader multidisciplinary technologies by the external research community. The XRF1 model can be released to research establishments under the terms and conditions of a Framework Non Disclosure Agreement (FNDA). The DLR used this model in several MDO related projects and the FP7 EU project Smart Fixed Wing Aircraft. A parameterization in CPACS format is available and could be used.

The pros and cons of using the XRF1 as a reference model are summarized below.

Pros:

1. experience at many research establishments
2. mature dynamic model
3. has been used also in FLEXOP

Cons:

1. NDA required
2. rules about IT security apply
3. restrictions for publication apply

CRM: NASA Common Research Model (B777 like)

The Common Research Model of NASA would be another choice for a reference aircraft model. The pros and cons related to this model are summarized in the following.

Pros:

1. free to use CAD
2. structural model available at DLR-AE (FERMAT config)
3. aero loft suitable for high fidelity CFD

Cons:

1. no CPACS available
2. not much experience with this configuration in the concortium
3. Boeing/NASA model

D150: DLR 150Pax Model (A320 like)

The D150 configuration was developed within the DLR project VAMP (Zill, et al., 2012). It is comparable to the Airbus A320-200. Data published by the manufacturer, for example on the Airbus website, and input data to the preliminary design program PrADO for the application example Airbus A320, are used for the D150 configuration (Klimmek, 2016). Its geometry is shown in Figure 449.

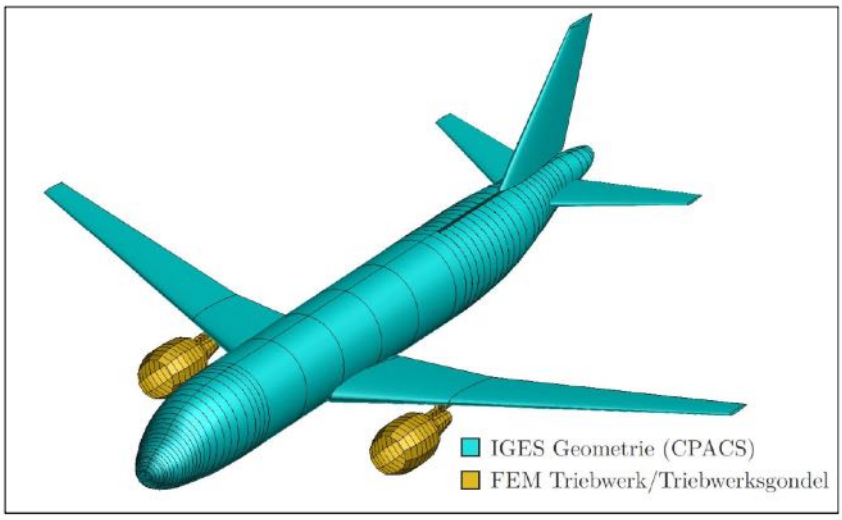


Figure 449: IGES-geometry of the D150-configuration

Table 87 lists the general parameters of the D150 configuration. The cruise speed V_C and cruise Mach number M_C are set to the maximum operational speeds V_{MO} and M_{MO}. The values for V_{MO} and M_{MO} for the Airbus A320 can be found in the EASA Type-Certificate Data Sheet (Frank, et al., 2012). The dive speed V_D can be calculated using the diagram of worksheet LTH BM 32 100-05 of the Luftfahrttechnischen Handbuch (LTH), and the dive Mach number $M_D = M_C + 0.07$ from the Acceptable Means of Compliance AMC 25.335(b)(**??**) of CS25.

The three airfoil profiles used for the four profile sections, with which the planform geometry is built, originate from the geometry of the DLR-F6 configuration. The DLR-F6 configuration is similar to the geometry of the Airbus A320 and was developed in the 1980s as a publicly-available geometry for aerodynamic studies.

Table 87: Main parameters of the D150-configuration

Wing	
Surface area	122.3 m^2
Span	33.91 m
Reference chord	4.19 m
Aspect ratio	9.4
Taper ratio	0.246
Sweep angle at 25% chord line	24.94°
HTP	
Area	30.98 m^2
Span	12.45 m
Aspect ratio	5.0
Taper ratio	0.33
Sweep angle at 25% chord line	28.0°
VTP	
Area	21.51 m^2
Span	5.87 m
Aspect ratio	1.6
Taper ratio	0.35
Sweep angle at 25% chord line	35.0°
Operational empty weight (OEM)	40638 kg
Maximum zero-fuel weight (MZFM)	60500 kg
Maximum take-off weight (MTOM)	72500 kg
Cruise Mach number	0.78
Cruise speed / Mach number	180 m/s EAS, Mach 0.82
Dive speed / Mach number	209 m/s EAS, Mach 0.89
Maximum flight level	12500 m

The pros and cons of using the D150 are as follows.

Pros:

1. DLR owned configuration
2. CPACS available

3. experience from multiple other projects
4. no publication issues
5. more relevant for industry

Cons:

1. short/medium range aircraft
2. potentially less benefits to demonstrate
3. aero loft not suitable for CFD

Reference Model Choice

At the time of writing of this report, the D150 appears to be the preferred choice for the scale-up task.

A final decision on this will be made by the consortium prior to starting WP4.

Workflow for the scale-up task (TUM, DLR, SZTAKI)

1. overall workflow (picture below, maybe make an updated version)
2. description of each block by partners

Simulink model data generation

The block ”Simulink Model Data Generation” receives Nastran decks from the ”cpacs-MONA” block and subsequently calculates the relevant aeroelastic model data by means of a Matlab based tool [Hofstee 2003]. The aeroelastic model data can then be forwarded and processed by the block ”Trim and linearize models”.

Trim and Linearize Models

This block receives aeroelastic model data from the ”Simulink Model Data Generation” block. It trims and linearizes for different operating conditions and fuel levels, in order to cover the flight envelope. For the ”Loads Analysis” block trim loads are provided. The ”GLA and MLA Control Synthesis” block handles the linearized models for the synthesis of GLA and MLA control laws.

Loads analysis The tool used for dynamic model generation and dynamic closed loop loads analysis is the DLR tool VarLoads. In a first step, the open loads loads are recalculated to ensure consistency with the previous sizing step. The loads are compared with the loads from the previous calculations with cpacs-MONA to confirm the stiffness and mass model.

The performance assessment of the GLA and MLA control relies on the loads, denoted as P_c, experienced by the aircraft structure during gust encounters. These loads are determined using the force summation method (FSM), expressed as

$$P_c = T_{cg}\left(P_g^{\text{ext}} - P_g^{\text{iner}}\right). \tag{75}$$

Here, P_g^{ext} and P_g^{iner} represent the external and inertial loads, respectively. The incremental loads from all load monitoring points are aggregated and transformed to the loads coordinate system using the transformation matrix T_{cg} [9, 50].

The gust input to the aerodynamic model consists of vertical 1-cosine gust profiles, characterized by the gust zone velocity $U_{z,t}(t)$ and acceleration $\dot{U}_{z,t}(t)$, defined as

$$U_{z,t}(t) = \begin{cases} \frac{\bar{U}_t}{2}\left(1 - \cos\left(\frac{\pi}{H_t}\left(U_\infty t - x_z\right)\right)\right), & \text{if } \frac{x_z}{U_\infty} \leq t \leq \frac{2H_t + x_z}{U_\infty} \\ 0, & \text{otherwise} \end{cases}$$
$$\dot{U}_{z,t}(t) = \begin{cases} \frac{\bar{U}_t \pi}{2H_t} U_\infty \sin\left(\frac{\pi}{H_t}\left(U_\infty t - x_z\right)\right), & \text{if } \frac{x_z}{U_\infty} \leq t \leq \frac{2H_t + x_z}{U_\infty} \\ 0, & \text{otherwise.} \end{cases} \tag{76}$$

Here, $\bar{U}t$ represents the maximum gust intensity, Ht is the gust half length, and x_z denotes the position along the aircraft [24]. The gust profile is traversed by the aircraft, as illustrated in Figure 144. Simulation of the 1-cosine gust encounters for the determined critical load cases provides the critical loads. Comparison of the calculations with the ones received from cpacs-MONA allows to validate the workflow up to this point.

For the manoeuvre loads the aircraft is trimmed for the most critical manoeuvres. The trim solution then provides the corresponding critical loads.

For the subsequent control synthesis tasks the open-loop nonlinear simulation model is trimmed and linearized to obtain states space models. Model order reduction is applied to these models to make them suitable for control law synthesis.

The flight points of the state-space models for the design of these load alleviation functions (LAF) are chosen according to the list of the computed load cases. The flight points for the active flutter suppression control laws are based on the 15% extended fight envelope required for certification of the aircraft according to CS 25 [24].

GLA Control Synthesis

The ”GLA and MLA Control Synthesis” block uses the set of state-space models for different flight conditions and mass cases.

The GLA control laws are designed for reduction of the wing root bending due to 1-cosine gusts. The control law is assumed to be a single gain matrix, which relates the onboard measurements to control surface deflections. In Figure 452 the control surfaces employed for GLA are framed in magenta. Each wing possesses ten aileron-like control surfaces. For simplification the control surfaces are grouped by two, which provides five control surfaces per wing. Furthermore, the elevator control surfaces are used for GLA as well. As vertical gust cases are considered exclusively, allocation of symmetrically corresponding control surfaces of the aircraft is possible. The output provided to the GLA controller is solely the z-acceleration from the fuselage IMU. Its orientation is depicted by the coordinate axes in Figure 452.

The GLA control law is determined by an optimisation. Based on a couple of gust encounters, covering gust half lengths of 30 ft to 350 ft, the maximum occurring wing root bending moment (WRBM) $P_{c,\mathrm{mx}}$ is determined. Thus, the main objective of the optimisation is to minimise the WRBM over all simulated gust encounters. Additionally, the control law should not cause to strong vibrations. Therefore, the maximum integral of the WRBM for all gust encounters should not exceed 90 % of the maximum integral of the WRBM in open-loop. Adding this constraint to the optimisation problem also prevents an unstable closed-loop behaviour.

Manoeuvre Load Alleviation

For certification of an aircraft, it has to be demonstrated that its structure can withstand the loads acting on it without damage. In order to design the structure accordingly, a so called loads envelope has to be computed. One of the critical load conditions comprising the loads envelope is the symmetrical pull up manoeuvre. For large transport aircraft, this condition is specified in the paragraphs CS 25.331 and CS 25.333 of the CS-25 issued by the EASA. One way of reducing the resulting wing root bending moment and hence the structural weight of the wing, is to shift the center of the lift distribution inboard by deflecting the control surfaces. This function is known as Manoeuvre Load Alleviation (MLA).

The objective is to find the optimal control surface scheduling to minimize the wing bending moment over the wing span. In particular the reduction of the integrated bending moment at the wing root will result in the most structural weight savings and is hence the indicator which control surface allocation to choose.

The aircraft model is trimmed for a pull-up manoeuvre with the resulting pitch rate to reach the desired load factor. The maximum deflections for the 10 wing trailing edge surfaces were set to be 10 deg upward. Such a setup requires a constraint optimization with the objective function minimizing the resulting wing root bending moment by symmetric deflection of the trailing edge control surfaces. Ultimately, to reduce the root bending moment the center of lift has to be shifted inboard. Physically, that means that the outboard lift is decreased, by upward deflection of the outer control surfaces.

Since the manoeuvre loads are computed by trimming the aircraft for specific flight condition like pull-up, push-over and roll manoeuvres, the design of a dynamic control manoeuvre load alleviation law is not necessary for the subsequent loads analysis. For the manoeuvre loads the allocation of the control surfaces however plays an essential role. To find the optimal spanwise cross-over location, where neutral position changes to full upward deflection of the trailing edge control surfaces, 11 discrete configurations were examined. The reference case 00 is without any MLA function, i.e. no deflection of the control surfaces. For configuration 01 all trailing edge surfaces are deflected upwards -10 deg. The configs 02 through 11 consecutively return more control surfaces from inboard to outboard to their neutral position, until for configuration 10 only the outermost aileron is deflected upwards.

The configuration 03, where all but the two innermost flaps are deflected upward is the configuration with the minimum wing root bending moment, as show in figure 453. The two flaps that are not deflected are inboard of the trapezoidal kink of the wing. The control surface layout is depicted in figure 452.

Please note that the sole selection criterion for the MLA control surface allocation is based on the resulting wing root bending moment. In the described workflow no other constraints have been considered. Such constraints could be for instance the trade off between wing loads and horizontal tail loads, buffeting constraints regarding the maximum lift coefficient at the dive Mach number, aerodynamic efficiency considerations of the control surfaces in a transonic flow regime, detrimental effects on a benign stall behavior, etc.

Active load alleviation benefits (SZTAKI)

It has been shown in [33] that for the D150 aircraft active load alleviation in the preliminary aircraft design stage can yield a significant reduction of wing box mass. The active load alleviation methods presented in [33] resulted in 6.2% less wing root bending moment using MLA and 10.7% less bending moment at the root of the active aircraft using GLA. This can lead to a wing structural mass reduction of 130.5 kg (0.18% of MTOM, 0.32% of OEM). The MLA and GLA methods presented in this report lead to approximately 20% decrease in the wing root bending moments with respect to the the open-loop gust loads. Based on these active load alleviation methods, the possible wing mass reduction is assumed to be approximately double of the amount of [33], which is a reduction of 260 kg. Therefore, the following values are used for the redesigned wing parameters, as given in Table 88.

The performance of utilizing active load alleviation methods is evaluated based on the range increase of the redesigned aircraft. This evaluation is done in the following way using the Breguet equation:

Table 88: Main parameters of the D150 baseline (reference) and redesign configuration

	Reference aircraft	Redesigned aircraft
Operational empty weight (OEM)	40638 kg	40508 kg
Maximum zero-fuel weight (MZFM)	60500 kg	60240 kg
Maximum take-off weight (MTOM)	72500 kg	72240 kg
Take-off fuel	10450 kg	10450 kg
Cruise Mach number	0.8	0.8

$$R = E\frac{V}{b_f g}\ln\frac{m_0}{m_0 - m_t} \tag{77}$$

where R is the range, V is the velocity of the aircraft, b_f is the thrust specific fuel consumption, $E = \frac{C_L}{C_D}$ is the lift to drag ratio, m_0 is the starting mass, m_t is the fuel mass consumed during the flight segment and g is the acceleration of gravity.

The parameters of the reference aircraft and the redesigned wing are given in Table 89. The range is evaluated assuming burning 90% of the total fuel.

Table 89: Main parameters and range results of the D150

	Reference aircraft	Redesigned aircraft
Take-off fuel	10450 kg	10450 kg
Take-off mass, m_0	71000 kg	70740 kg
Landing mass (10% fuel remaining)	61595 kg	61335 kg
Fuel consumed, m_t	9405 kg	9405 kg
Cruise Mach number	0.8	0.8

The Breguet equation indicates that with active load alleviation, the range of the D150 can be extended by 0.4%. If the range is to be kept as for the initial aircraft configuration, then the saved wing mass can be utilized for more passenger seats.

Wing shape control benefits (SZTAKI)

The drag of an aircraft can be optimized via active wing shape control. This is achieved by utilizing the ailerons for trimming the aircraft for various aspect ratio and mass cases.

Drag estimation using the PANUKL code

PANUKL is a software package to compute the aerodynamic characteristics of an aircraft using low order panel methods [30]. 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 as shown in Figure 454.

Computational method of PANUKL

The 3D Panel method creates thick surfaces for modeling the aircraft body. As mentioned in the latter section, the panel elements represent the flow only and not the actual object. On each rectangular surface, one doublet sheet and one source sheet is placed. This method does not include vortices on the body, due to the fact, that a rectangular doublet sheet is the equivalent of a ring vortex. The flat vortex wake during simulations is assumed to be parallel to the chord and originates from the trailing edge points. Note that PANUKL offers a second option for the setting of vortex wake direction, in which the wake is parallel to the angle of attack. A series of testing led to the conclusion that the first option is more practical and precise.

The method's prime target is to solve the Laplace equation for the full velocity potential

$$\nabla^2 \Phi_{velo} = 0 \tag{78}$$

The full velocity potential is defined as

$$\Phi_{velo}(x,y,z) = \frac{1}{4\pi} \int\limits_{Body+Wake} \mu \frac{\partial}{\partial n}\left(\frac{1}{r}\right) dS - \frac{1}{4\pi} \int\limits_{Body} \sigma \left(\frac{1}{r}\right) dS + \Phi_{\infty} \tag{79}$$

The following three boundary conditions are assumed.

- The Dirichlet boundary condition specifies the value of the potential function inside each panel

$$\frac{1}{4\pi} \int\limits_{Body+Wake} \mu \frac{\partial}{\partial n}\left(\frac{1}{r}\right) dS - \frac{1}{4\pi} \int\limits_{Body} \sigma \left(\frac{1}{r}\right) dS = 0 \tag{80}$$

  where
  - Doublet strength: $\mu = -(\Phi_{velo} - \Phi_i) = -\varphi$
  - Source strength: $\sigma = \frac{\partial \mu}{\partial n}$

- Kutta-Joukowsky condition on the trailing edge

$$\Delta p(x,y)_{TE} = 0 \tag{81}$$

- On the vortex wake

$$\frac{\partial \varphi(x,y)}{\partial x} = 0 \tag{82}$$

The aircraft's body surface is approximated by flat panels, which allows to approximate Equation 80 with a set of linear algebraic equations containing unknown doublet strength that is constant for each panel

$$\sum_{k=1}^{N} C_k \mu_k + \sum_{l=1}^{N_\omega} C_l \mu_l + \sum_{k=1}^{N} B_k \sigma_k = 0 \tag{83}$$

where C_k, C_l and B_k denote influence coefficients

$$C_k = \frac{1}{4\pi} \int_{S_{1234}} \frac{\partial}{\partial n} \left(\frac{1}{r_k} \right) dS_k; \tag{84}$$

$$C_l = \frac{1}{4\pi} \int_{S_{1234}} \frac{\partial}{\partial n} \left(\frac{1}{r_l} \right) dS_l; \tag{85}$$

$$B_k = -\frac{1}{4\pi} \int_{S_{1234}} \frac{1}{r_k} dS_k \tag{86}$$

The indexes in Equations 84 - 86 represent

N = number of panels on the aircraft surface

N_w = number of panels on the wake

S_{1234} = area of the k^{th} panel

The source strength σ, also constant for each panel, can be defined as the main component of the radial normal velocity on each panel

$$\sigma = -n \cdot V_\infty$$

The resulting set of equations will have the doublet strength as the unknown. To determine it on the vortex wake (W), the Kutta-Joukowsky boundary condition is applied

$$\mu_{TE} = \mu_W = const$$

The doublet strength on the wake is equal to the difference between doublet strength on the upper (U) and lower (L) side close to trailing edge (TE) and can be expressed as:

$$\mu_W = \mu_U - \mu_L$$

This completes the set of equations and only the integrals in Equations 85 - 86 have to be determined, resulting in the velocity potential distribution on the body surface. At this point, the previously defined full velocity potential in Equation 79 can be obtained.

Determination of important coefficients

The output of the aerodynamic analysis, among others, includes the loads acting on the aircraft's surface, which is then used to calculate the deformation. In Figure 458 a simplified flowchart is seen showing the steps involved in the determination of force and moment components.

As seen in Figure 458, the gradient of the velocity potential results in the velocity distribution, which can be achieved by differentiation of Φ_{velo} with respect to defined tangential coordinates. Due the irrotational

nature of fluid flow, Bernoulli's theorem states that $p + \frac{1}{2}\rho V^2 = const.$ between any two points on the flow. With the application of this, the pressure distribution can be derived.

The main load components can be obtained from the pressure distribution by integrating over the panels.

Lift force:

$$F_Z = -\sum_{i=1}^{N} p_i S_i \mathbf{n_i} \cdot \mathbf{z} \tag{87}$$

Drag force:

$$F_D = -\sum_{i=1}^{N} p_i S_i \mathbf{n_i} \cdot \mathbf{x} \tag{88}$$

Pitching moment:

$$M_y = \sum_{i=1}^{N} p_i S_i x_i \mathbf{n_i} \cdot \mathbf{z} + \sum_{i=1}^{N} p_i S_i z_i \mathbf{n_i} \cdot \mathbf{x} \tag{89}$$

PANUKL uses a Trefftz-plane approach for calculating the induced drag coefficient.

PANUKL results for the D150 aircraft

The goal is to use PANUKL to obtain the induced drag values for the reference aircraft (with zero aileron deflections) and the minimal induced drag corresponding to the optimal aileron configuration. It is assumed that the aircraft has 10 ailerons as shown in Figure 459.

PANUKL is configured in the following way:

- Mach: 0,8; Altitude: 10 km
- 2800 simulations: Investigated angles [-1...1,5] degrees
- Outputs: C_{Di} and C_L values for each aileron combination

PANUKL results:

- C_{Di} for the reference case: 2.705×10^{-3}
- C_{Di} for the optimal aileron setup: 2.543×10^{-3}

The optimal aileron deflections are given in Table 90.

Table 90: Optimal aileron deflections

Ailerons	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
degrees	0	1	1.5	1	1	2	1	2	0.5	0

Substituting the results of the optimized induced drag and reference induced drag models, the Breguet equation indicates 6.37% range increase compared to the reference aircraft model. Note that since induced drag values are substituted into the Breguet equation instead of the drag coefficient, the range increase using optimal aileron configuration can lead to a smaller increase in real implementation.

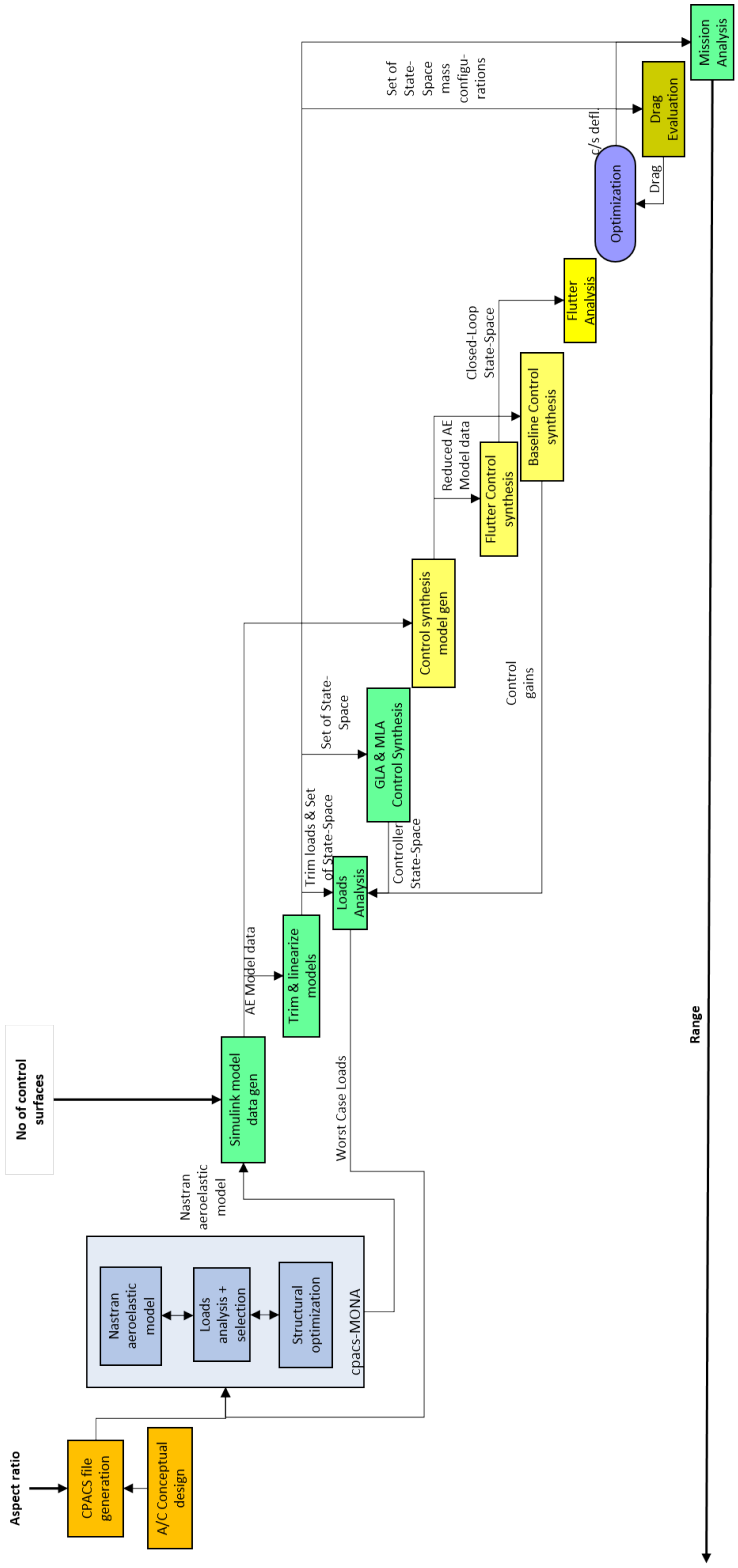


Figure 450: Scale-up workflow.

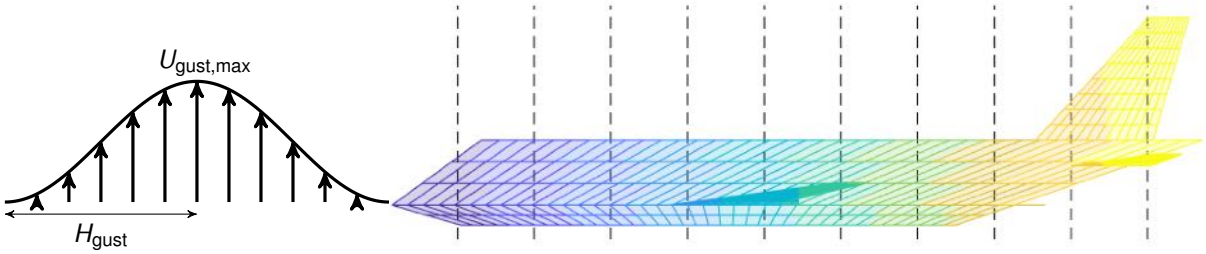


Figure 451: 1-cosine gust and aircraft gust zones.

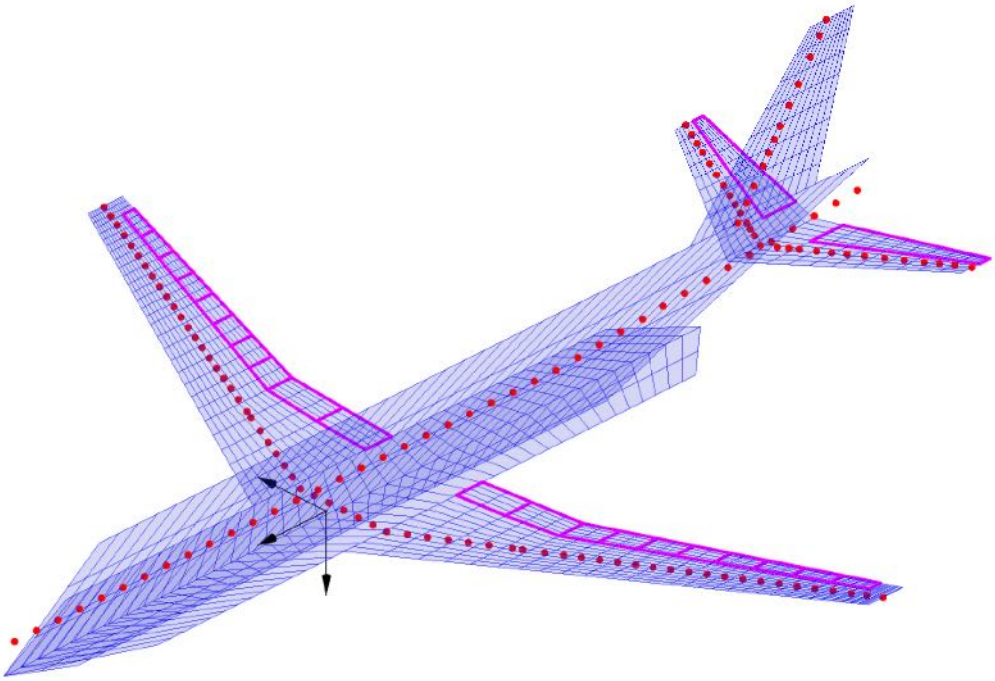


Figure 452: D150 flexible aircraft model defined by the structural grid (red), the aerodynamic panel model (blue), the deployed control surfaces for GLA (magenta), and the IMU coordinate system location and orientation (black).

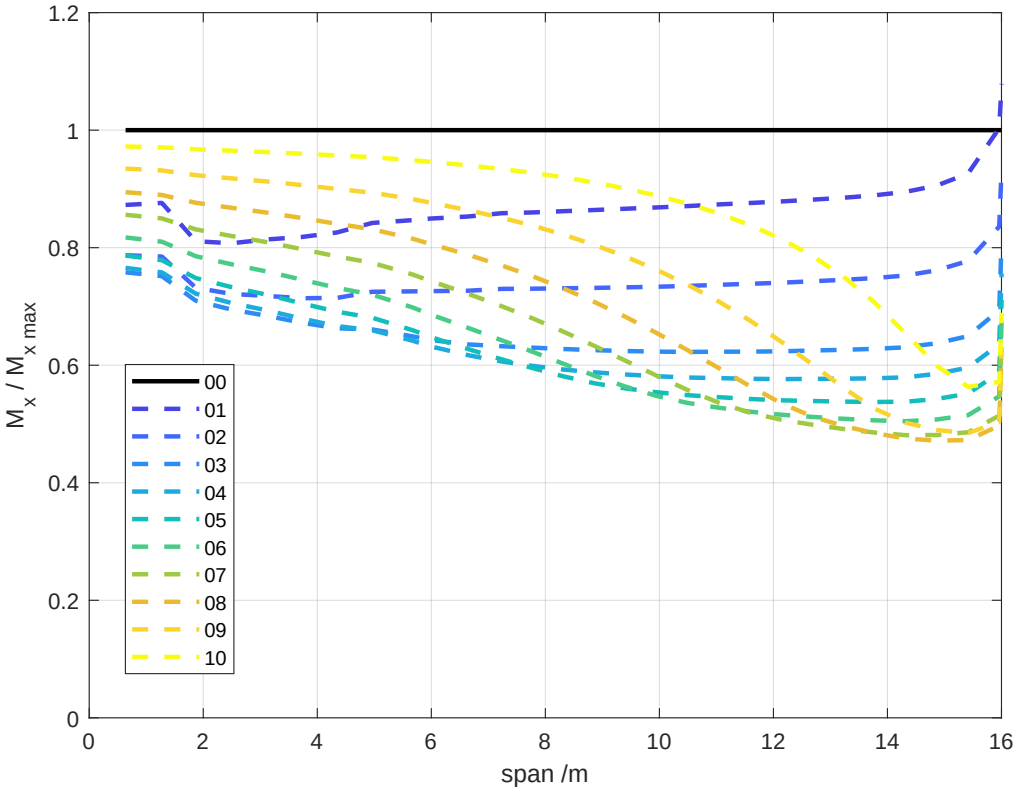


Figure 453: wing bending moment distribution of the various MLA control surface configurations normalized by the unreduced bending moment distribution.

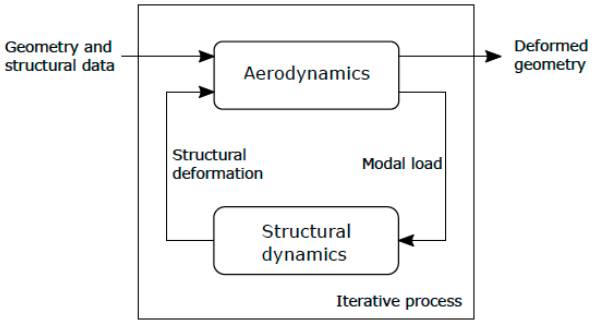


Figure 454: Trim flight deformation calculation process

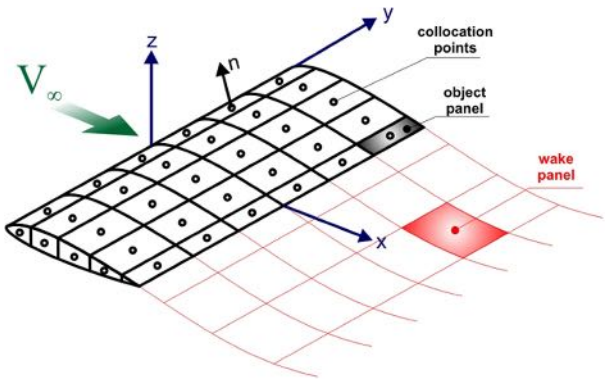


Figure 455: Approximation of the body surface by panel elements

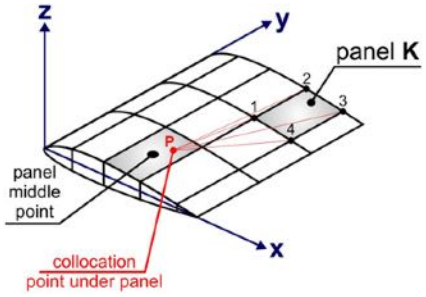


Figure 456: Influence of point P on k^{th} panel

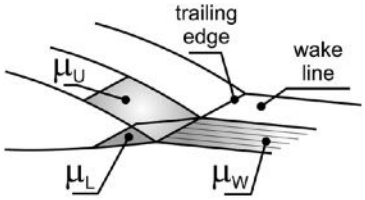


Figure 457: Connection between doublet strength on the trailing edge and wake

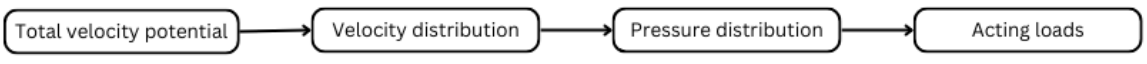


Figure 458: Flowchart of loads calculation with the PANUKL based tool

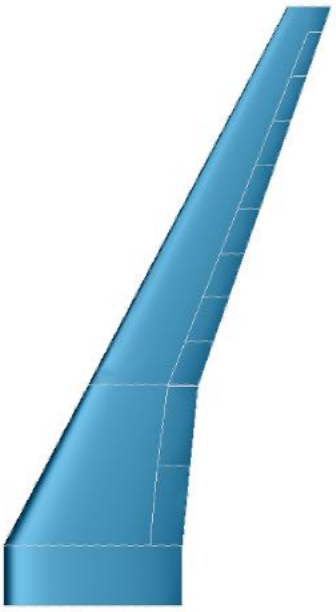


Figure 459: 10 aileron configuration of the D150 aircraft

2.4.4 Deviations, their reason, impact on the project and corrective actions

2.5 Explanation of the work carried out per WP - Work Package 5

2.5.1 Objectives and activities

- Project Coordination
- Evaluation of collaborative tools and their best practices
- Management of exploitation and dissemination of project results

2.5.2 Starting point and approach

The consortium is made up of four beneficiaries. Three of them (SZTAKI, TUM, DLR) have been involved in our previous coordinated project (FLEXOP) and with the fourth one (ONERA) we have had several common H2020 projects already (VISION). The previous cooperations resulted in a smooth project implementation on the management side.

2.5.3 Efforts and achieved results, name involved contractors

Task 5.1: Project Management (SZTAKI)

The main activities of the Management Team were:

- Ensured achievements of overall project schedule and objectives by

  Constant monitoring of project achievements against the work plan – there was a notable delay in project implementation due to

- the outbreak of Covid-19 epidemics. Some partners could work remotely but others were obstacled by the restriction that they could not visit their permits/laboratories. And also the lack of in-person consortium meetings made the implementation more complicated.
- chip shortage – procurements were delayed by chip shortage. We had to redesign the flight control computer, we had to reconsider procurements and redesign printed circuit boards because of this shortage.
- rescheduling the project after the aircraft accident. The resources that were planned for advanced wing had to be reallocated for rebuilding the aircraft, and also in general the timeline for rebulding the aircraft was really tight.
- changes in airfield and the issuing of the light permits. There was unplanned workload due to changes in flight permit issuing. We also had to change the airport because of these changes – we moved all flight tests to Cochstedt. We always received the flight permit only last minute which made organization difficult and costly. We also had fixed two weeks window for flight test because we had to move to the plot while Oberpfaffenhofen would have been easier as it is close- does not need so much preparation. In addition statistically Cochstedt is a much windier airport which caused some cancellations.
- Fluctation in key personnel: (examples) DLR – MDO expert leaving the company, TUM - Technician was leaving in a critical moment – training of new people was consuming time and other resources (human resources and budgetwise)

  o Identification of risks and definition of risk mitigation measures through the Risk Reigster

o Solving any technical, financial, administrative or contractual issues or conflicts between partners, when needed – an amendment with project extension and budget reallocation was implemented in the second period of the project:

AMENDMENT:

- 1. Changes of Annex 1 (description of the action)
- 2. Change of the action's duration
- 3. Change of reporting periods
- 4. Changes of Annex 2 (estimated budget of the action)

These change(s) were necessary for the following reason(s): On 30 August 2022, FLIPASED's T-FLEX remotely piloted experimental demonstrator suffered an in-flight accident at DLR's Cochstedt Flight Test Center. The investigation pointed to a loss of radio control, an off the shelf equipment that unexpectedly failed. The encountered anomaly caused an impact to the ground, which damaged the demonstrator and stopped immediately the flight test campaing. Early September 2022, the consortium started to repair the T-FLEX to be able to finish the testing activities. However, the repair works have not been concluded in time due to the delays in obtaining replacement parts and in addition the wheather testing window was soon gone. As a consequence the finalisation of the demonstration was not performed until early Spring 2023. The extension did not impact the total costs and EU contribution for the project. The consortium commited to conclude all its work by M46. Therefore, the coordinator requested to extend the project duration by 6 months (from 40 months to 46 months) to have all the activities finished by the end of June 2023. On the HR side, the partners replanned the resources to account for the 6 months extension. Due to staffing commitments, ONERA was not able to re-allocate significant effort to 2023. The other project partners have taken over most of ONERA's commitments, hence no extra funding was needed. In practice, part of ONERA's remaining resources were mainly reallocated to TUM to cover the flight tests. In addition some other direct costs were transferred to personnel costs within each beneficiary. In terms of effort, ONERA decreased it from 58PM to 35PM while the rest of partners increased their PM between 12% to 84% thanks to the ONERA budget reallocation and to their Other direct costs conversion into personnel cost.

• Handled and distributed the funds according to the rules agreed within the consortium – pre-financing was distributed according to the Consortium Agreement.

1	SZTAKI	800 156,25
2	TUM	926 531,25
3	DLR	706 121,25
4	ONERA	451 875,00

In the second period the interim payment was distributed among partners based on their eligible costs declared for the first period.

1	SZTAKI	106 687,500
2	TUM	123 537,505
3	DLR	94 149,505
4	ONERA	60 250,000

• Maintained regular contact with the partner organisations

• Established a scientific and industrial advisory group

- Prof. Peter Seiler, Faculty of Electrical Engineering & Computer Science, University of Michigan
- Daniel Ossmann from Munich University of Applied Sciences
- Roeland de Breuker from Technical University of Delft.

  The Scientific Advisory Group (SAG) was confirmed by the Steering Committe in the first month of the Project for the purpose of offering advice and support on a wide range of Project-relevant issues. Members of the SAG are internationally recognised experts in the field of the Project.

  The Industrial Advisory Group (IAG) includes key experts from the FLiPASED domains representing the key OEMs from Europe. Members were confirmed by the Steering Committe in the first month of the Project

- Sebastien Blanc A350XWB loads and aeroelastics Designated Expert and Airbus - Active Adaptive Wing Leader, Airbus Commercial Aircraft
- Carlo Aquilini, Airbus Defence and Space
- Colo Ludovic (Aero-Structural design directorate) From Dassult- Aviation
- Olivier Cantinaud (Technical Systems Directorate, Flight Dynamics Department) From DassultAviation

Both IAG and SAG members were invited to several project meetings online and in-person as well. And whenever the project reached a critical issue or milestone they were consulted for their professional advices.

• Managed risk and settle any disputes within the consortium

• Organised the management team meetings, consortium meetings and meetings with scientific advisory group

Kick-off meeting: of the project has taken place at (and hosted by) SZTAKI on 12-13th September 2020. All 4 of the partners and the members of the Scientific and the Industrial Advisory Group got together for the first time. The meeting started with presentations of each partner and followed by project presentations. Steering committee meeting was also held where the members of the Management Support Team and the Scientific and the Industrial Advisory Group as well as the WP Leaders were elected.

1st Progress Meeting: of the project was planned to be held on the 19-20th of March in Münich but it was postponed and replaced by several thematic (and WP specialized) online meetings due to the emerging COVID situation.

The actual first progress meeting was organized on the 13-16th of November 2020 (Friday-Monday) where the progress of tasks was discussed and rescheduled in detail. See the recovery plan in section Deviation.

See the meeting folder with relevant presentations and minutes:

`https://dms.sztaki.hu/nextcloud/s/s8n79K4HJPMTQbp/download`

First in-person meeting at TUM after COVID travel restrictions was held on 2021/10/15 in Münich. After 20 months of online work (due to COVID-related travel restrictions) our consortium has finally met in person at TUM Münich. We have updated the wing design objectives for the aeroservoelastic design toolchain**.** Based on the new performance criteria for MDO optimisation we refined the objectives of the flight test campaign. The basis for scale-up (D150) and the corresponding tools have been discussed. TUM showed the 50% scaled down version of the demonstrator called: DeFStaR – with which they performed stall tests.

DirectDrive integration at TUM SZTAKI team visited TUM in November 2021 in order to integrate the DirectDrive actuator to the T-FLEX aircraft's flexible wing and perform regular flight control computer updates. Because of the pandemic, we ensured a safe work environment with daily rapid testing at TUM facilities and by using FFP2 masks in the lab. We have successfully tested SZTAKI's custom-developed servo actuator which will be responsible for flutter suppression using the outermost ailerons.

Aircraft construction and reconstruction activities, GVT and flight test preparation activities meant numerous travels for whole teams visiting construction and testing sites. For further information on these meetings, campaigns and travels please see WP3 description.

3rd Consortium meeting at Budapest: After the long period of ONLINE MEETINGS only the whole FLIPASED consortium could finally meet in person on the 4th and 5th of July 2022 in Budapest, at SZTAKI in hybrid format so also the members of the Scientific and Industrial Advisory Group could join us for the event on the 5th of July. The 4th of July was dedicated to bilateral in-person meetings. Progress was achieved regarding the following topics:

- Analysis of the recent flight test campaign at DLR Cochstedt
- Preparation for the manufacturing and instrumentation of the newly built performance adaptive wing
- Schedule and objectives for the upcoming GVT (Ground Vibration Test) campaign, to be conducted at DLR in Göttingen
- Target for the next flight test campaign and the corresponding software, hardware infrastructure to be provided

4th Consortium meeting at Budapest A new consortium meeting of the FLIPASED project was organized in Budapest on October 11 and 12 2022. Most of the partners were able to come to Budapest in person and the remaining researchers joined us online. There were two main topics for discussion, the detailed rebuild and flight test plan of the demonstrator aircraft and the status and future goals of the multidisciplinary design optimization (MDO) toolchain. Progress was made in the following areas:

- Detailed airframe rebuilt plan in a Gannt chart from, specifying targets, deadlines and responsibilities for all partners
- Improvements in Hardware in the loop (HIL) tests in order to decrease the technology gap between the HIL tests and flight tests of the demonstrator aircraft
- Updated ground vibration tests (GVT) plans
- Flight test plan specifications for the active flutter suppression demonstration tests
- MDO tasks for the demonstrator and scale-up aircraft

FINAL REVIEW IN BRUSSELS The consortium organized a rehearsal day on the 11th of July 2023. The final review took place on the 12th of July 2023 at DLR office, Brussels. All consortium members participated and the WP leaders successfully presented the progress made during the project as well as the results achieved.

Final Meeting in Budapest: took place after the project's end date. It was organized in Budapest on the 20th and 21st of July. Organizing a meeting after project closure that is funded from the project members' own resources clearly shows the engagement of the partners. Flight tests were conducted and concluded only short before project end so further work and discussions were necessary in order to reach the goals of the project. Also further project dissemination and scientific publications had to be planned.

weekly webexes were held by the coordinator – usually dedicated to a WP or a relevant deliverable – 40 meetings until the end of the first period and further 73 meetings in the second period.

WP/deliverable webexes were also organized by the coordinator or by different partner organisations whenever needed by the workflow – 10 meetings (1st period) and more than 15 meetings (2nd period) were dedicated to Fligt Testing and flight test data processing (3), to D1.2 Toolchain workshop, but also to Flutter Evasive Action, to MDO and also to different deliverables.

See the meeting folder with relevant presentations and minutes for all the meetings:

`https://dms.sztaki.hu/nextcloud/s/s8n79K4HJPMTQbp`

• Reported to and chaired steering committee on the consortium meetings

• Reviewed and validated the project reports to ensure consistency with the project tasks (especially in the case of reviewing the different project implementation concepts and deliverables)

• Submitted reports and other deliverables to the Commission – all deliverables have been submitted although there were some deviations regarding their content described in the workplan – please see the relevant WP descriptions regarding these deviations.

• Transmitted documents and information connected with the Project to and between the Work Package Leaders and the partner concerned

• Prepared and updated the schedules of the whole project whenever needed.

• Ethical, social and gender issues encountered during the project life were monitored. It includes activities for preparing the gender issues plan and support to the other partners for applying the plan. (During Grant Preparation phase a separate deliverable was introduced to the project in a new WP called EPQ - Requirement No. 1 – the deliverable was submitted in due time). Referring to the Horizon Europe requirements SZTAKI has already developed its Gender Equality Plan and stareted its first related activities by setting up the internal council (implementing the plan) and by organising the first gender workshops within the institute.

A Project Handbook defining procedures, templates and methods for the assessment of project achievements was issued in the beginning of the project. It was also submitted as a deliverable.

The project webpage was set up and also the requested data management plan was submitted on due time.

At month 12 and at month 24 two progress reports on project level were issued indicating the status of the project.

The organisation of the workshops with the scientific advisory group were financially supported.

The travel expenses of the scientific advisory group were financially covered by WP5.

Task 5.2: Collaboration tools, methods and practices (SZTAKI)

Common problem in multidisciplinary projects is the lack of understanding between partners due to their background and expertise, which leads to conservative designs or creates miscommunication, risking delays, costly re-designs or redundant solutions for the same problem by multiple stakeholders. We planned to tackle these issues by implementing collaborative project management solutions. After a thorough analysis of the different workflows and work groups we decided to use the following tools: Nextcloud for sharing, editing documents, defining tasks. Webex for online meetings. overleaf and GIT.

The consortium established collaborative tools for project management (Nextcloud + Agantty), software development (Git), document editing (Overleaf). Moreover the collaborative work process also involves common hardware development tools - a common hardware-in-the-loop platform. The partner contributions within the common MDO toolchain are all implemented and tested using the RCE environment.

Goal: tackling common problems of multidisciplinary projects

Achievements: The consortium established collaborative tools for project management (Nextcloud), software development (Git), document editing (Overleaf). Moreover the collaborative work process also involves common hardware development tools - a common hardware-in-the-loop platform. The partner contributions within the common MDO toolchain are all implemented and tested using the RCE environment. Matlab based flight test data analysis and updating workflow.

Task 5.3: Exploitation and Dissemination Management (SZTAKI)

This task includes:

• Observation of the evolving research and development trends as well as communication of the observances to the consortium members – follow-up done by the coordinator.

• Co-ordination of issues related to Intellectual Property Rights – this topic is regulated in the Consortium Agreement the partners have signed.

• Set-up of an Exploitation and Dissemination Plan; dissemination of results was achieved by publications of individual partners. Furthermore a panel session was dedicated to our project on AIAA Scitech, organised in the USA, National Harbor (for further details see D5.7) A significant amount of publications have been submitted by project partners (see in Part A data on the Particiapnt Portal). The Exploitation and Dissemintation Plan was set up by the 31st of August 2021 and also The Exploitation and Dissemintation report has been submitted at the end of the project.

• In accordance with the dissemination plan the consortium members had to identify results with potential for patenting and publication activities had to be aligned with patent application rules – this topic is regulated in the Consortium Agreement the partners have signed.

2.5.4 Deviations, their reason, impact on the project and corrective actions

See section Deviations of this Deliverable.

2.6 Impact

2.6.1 Project impacts on users or the results

It is foreseen that several research organizations and aerospace companies will benefit from the project results. The project showcased active flutter control results and the corresponding design tools with both of them shared open access. This will most likely trigger an interest in not just only reading the corresponding reports and publications but users of the results will have the opportunity to analyze the results in detail. The plan of the consortium is to disseminate the results even after the end of the project: several presentations will go in details of the results during the IFASD 2024 conference and people will be encouraged to test the results. It is also the plan of the consortium to present the results in various other project context.

2.6.2 Project impacts on the aviation research community

The project showed the conceptual design tradeoffs and the drag reduction potential of an advanced high aspect ratio wing aircraft. These results do not exist on their own they are part of a larger puzzle providing technological background for future generation of aircraft. On the other hand the approach the consortium took, to include flight controls within the MDO loop is fairly novel and both DLR and ONERA are working on follow up actions to this approach. On the other side of the Atlantic University of Michigan and NASA has similar plans, hence showing a feasible strategy and sharing the initial results will fuel further research actions along this topic.

The success of active flutter suppression on a conventional configuration flying vehicle is a major milestone in aviation research and by maturing the TRL level of this technology several OEMs might be closer to introduce similar technology on their future aircraft. Consortium members are already in discussion with several OEMs, and it is the aim of the consortium to brief both EASA and FAA about the potential use of the technology.

2.6.3 Project impacts on partners future endeavors

Both ONERA and DLR are already working on extending the project results in national and Clean Aviation projects. Project results and experience will serve as a solid foundation to continue research and innovation in the fields related to FliPASED. TUM succesfully proved and gained significant expertise in flying large UAVs and hence future endeavors might lead to building and flying similar size vehicles within new national and EU projects. Moreover project members gained significant experience not only with flight testing but also with aeroelasticity, flight controls, avionics and mathematical modelling on several levels what will be included in the curriculum of the university courses. The methods and tools, as well as the experience flight testing advanced active control methods will help SZTAKI work on National and international projects related to aeroservoelasticity with both academic and industrial partners. This might include not only aeronautics but space related projects, pseudo satellites and control development for integrated elastic response control of future high aspect ratio aircraft.

2.6.4 Competition analysis

In the MDO field both DLR and ONERA are very active with other groups not present within the project, hence even an internal competition is present for the partners. Other universities and research centers working on the topic include Supaero, University of Michigan, NASA, Stanford University, Politechnico Milano (NeoCASS Suite), Imperial College (SHARPy), Technion with the Modal Rotation Method (MRM), TU-Delft PROTEUS, Bombardier in-house tool, Gulfstream ATLASS, EMWET, and many others. Some of them already started looking at controls, but mostly these tools are able to take into consideration only static flap deflections (for maneuver loads), but dynamic control action is not considered in them - unlike in FliPASED where the maneuver envelope is completely shifted by taking credit of the fully dynamic MLA and GLA functions.

On the active flutter control field the notable competition are NASA (X-56 program) and the related PAAW (NASA funded performance adaptive aeroelastic wing program). In europe Technion is working on the A3TB flutter demonstrator and Politechnico Milano (together with University of Washington) works on an active flutter suppression wind tunnel model. The flight test results of these projects all aim at demonstrating flutter suppresion on a blended wing body aircraft, where the first flutter mode is almost always the body-freedom-flutter (rigid body pitch and wing torsion coupling), what is a lower frequency phenomena than the pure wing bending-torsion coupling present on P-FLEX and most likely also one of the dominant flutter modes foreseen on high aspect ratio commercial aircraft of the near future. Dassault Aviation, within the Concerto Clean Aviation program is also planning to flight test a similar configuration demonstrator to P-FLEX, but that will most likely occur at least 3 years after the success of FliPASED.

2.6.5 Intellectual property

The project partners did not file any patents during the project and are not planning to do so. The real intellectual property generated within the project is spread among multiple disciplines and fileds and very difficult to cut out certain results what could be individually protected.

On the other hand every partner matured its tools and methods or even developed components (like operational modal analysis computer, direct drive actuator, ground control station infrastructure, etc.) where the gained intellectual property is captured and retained for future use.

2.6.6 Dissemination and communication activities

Impact of the project is established by the conference and journal publications of the partners, but this is far from enough in the present day. Impact is multiplied by the social media appearances, television interviews and in-person person presentations to key aviation stakeholders. Details of the corresponding dissemination activities can be found in D5.7.

3 Update of the plan for exploitation and dissemination of result plan

The achieved dissemination and exploitation results and activities within the 46 months duration of the project were very strong. The core objective of WP5 besides everyday project management of the consortium members was to disseminate key findings and outcomes of the project in a structured manner in order to maximise project impact and outreach across key stakeholder groups. As outlined in the FLiPASED description of work, the dissemination objectives are to: • Identify the main dissemination target groups and ensure the adequate promotion of the project, its activities and results, • Prepare materials for the dissemination activities, • Maximise the dissemination potential of the project outputs to the aerospace community, for the members of parallel H2020, Horizon Europe and Clean Aviation projects, • Provide a plan for exploitation of the project outputs and support their long-term effects, • Organise a final workshop for the presentation of the project results and support information to the advisory group. The dissemination of FLiPASED has been essential throughout the project's life and needed to be carried out with the cooperation of all work packages and all project partners. The aim of this document is to provide the dissemination, communication and exploitation activities as well as the impact of these actions to fulfil the objectives of WP5 described in the FLiPASED GA. This deliverable will show the achievements of WP5, i.e.: • The development of all planned dissemination tools, • The creation of all planned publications (project brochure and newsletters), • The use of social media to communicate efficiently on the project; • The complete list of disseminated FLiPASED activities at events such as workshops, conferences, webinars and internal meetings, • The cooperation with other H2020 and EU funded projects, and • The organisation of a final workshop event, It will also detail the exploitation measures that have been undertaken during the 46 months duration of the project and will present the exploitation plan of the project partners after project end.

The FliPASED team spent significant effort in disseminating the project results and foresees a great exploitation potential in the future both to the general public as well as to the professional aerospace research and development community. Several lists of relevant documents, publications and other key references are presented within the deliverable D5.7 abou the details. The key exploitation results and short-term targets are also discussed within that document. The numbers, even though many of them are estimates, show significant outreach and showcase a highly successful project reaching target audience.

4 Update of the data management plan

The data management plan as described in the DoA needs no update.

5 Follow-up of recommendations and comments from previous review

Corrective actions recommended by the project officer and external reviewer at the first periodic review and the response of the consortium to them (M20):

For WP1, the deliverable D1.5 should be revised to clearly state that the decision for the D150 reference model has been taken following an impact driven principle as the Next Generation Single Aisle is the most probable new product development for OEMs. Then it is also expected to include in the revised version of D1.5 which aspects of the other WPs are affected by the Single Aisle decision in relation to a Wide Body reference aircraft.	D1.5 was ravised based on the input from the review – more emphasis was made to draw the connection with the Next Generation Single Aisle efforts
For WP2, as consequence of the delays, the consortium should evaluate the alternative solution of using a tool-chain based adaptation on one of the already existing wings. Since the goal is to have more control surfaces instead of creating a completely new clean design, the consortium must reassess the manufacturing of the planned -3 wing. Indeed, this decision is critical for the schedule of the project and has to be taken based on its potential impact to achieve the objectives.	Based on the suggestion the consortium reassess the manufacturing of the planned -3 wing from clean sheet and instead retrofitted the -0 wing
For WP3, the pending deliverable D3.2 should also include the justification of changes in the landing gear of the demonstrator due to safety concerns with enough level of detail. Other modifications that have delayed this WP, such as the power issues, will have to be considered as well.	The deliverable was revised and the landing gear and safety related modifications were discussed in more detail
For WP4, the scale up activities should be developed in parallel to flight testing due to the project time constraints. Therefore, some of the scale up activities will be performed on items that may not be fully flight tested validated. This must be explained and reported in D4.3.	The team took the advice and worked on two parallel tracks slightly de-coupling the scale-up and the flight testing, D4.3 will be revised and the reasons will be ellaborated in it
The FLIPASED consortium is determined to disseminate and communicate the project activities and results by other means than scientific publications. The efforts to fulfil the dissemination and communication obligations should be significantly increased during the second phase of the project.	Several outreach and dissemination activities were done by the consortium to publicize the results
In order to ensure the interest of industry, more frequent and efficient meetings with the Industrial Advisory Group (IAG) should take place.	Both in-person as well as email communication was initiated with the SAG members during consortium meetings, conferences and after major milestones of the project

General comments by the project officer and external reviewer at the first periodic review and the response of the consortium to them (M20):

In terms of the impact potential, FLIPASED's High Level Objectives (HLO) aim at 10% fuel efficiency improvement, 20% gust peak amplitude reduction, 50% models for certification reduction. The submitted reporting documents show 3% reduction with respect to pure drag reduction control law alone, without any modification (re-design) of the wing. The goal of 10% drag reduction is assessed as achievable with new wing design in addition to active and passive load alleviation and drag reduction functions. For peak gust amplitude reduction, the preliminary results on the existing model show 12% and 18% reduction in the positive and negative part of the response respectively, but final analysis with the re-designed wing has not been performed yet.	Final results of the scale-up have not been documented yet: Peak gust amplitude reduction is subject to several factors – sensors, actuators, wing design etc. – certainly we are able to show a case where we achieve that but the interaction of different disciplines is difficult to asses The 10% fuel efficiency improvement is achieavable with higher AR wing + lighter wing structure (enabled by MLE,GLE,AFC) + flight envelope dependent trailing edge scheduling (with high nuber of flaps)

6 Deviations from Annex 1

6.1 Tasks

6.1.1 WP1

The main deviations arose in WP1 as already described in Chapter 2.1.3. Since the first part of the project was heavily impacted by COVID the partners could not spend the necessary time together to establish the model setup, integration and overall workflow. Moreover one of the key person establishing the workflow left DLR and it was difficult to find a suitable replacement for him.

The original intention was also to have a tighter conncetion between the demonstrator RCE workflow and the flight tests of the demonstrator, but as both schedules slipped the coupling became looser.

6.1.2 WP2

The main deviations arose in WP2 as already described in Chapter 2.2.3. This workpackage includes development, maturation and refinement of various tools required for flight testing the flutter as well as the active wingshape optimized demonstrator and also includes development of the necessary tools to run the MDO toolchain. One of the main deviation arised from the fact that significantly more iterations were required to come up with the baseline demonstrator model, hence several of the tasks dependent on it has to be iterated a number of times or were started with a delay. The numerical stability and sensitivity of the solutions to small changes in the models also led to significantly more effort and time spent on control design tasks, especially the baseline controller. The original intention was to flight test and refine several of the technologies, but due to the slipped schedule and the difficulties with the demonstrator, some of the methods have been evaluated in sumulation only.

6.1.3 WP3

The main deviations arose in WP3 as already described in Chapter 2.3.3. At the start of the project many problems arose: • Landing gear and ground control issues • Engine issues • Aerodynamics not as predicted with significantly less lift • COVID-19 restrictions affecting workshop and flight testing These challenges proved difficult to overcome, resulting in only two flights during test phase 1. The most significant issue was the delayed resolution of the aircraft's ground controllability problem. This delay stemmed from restricted workshop access due to COVID-19 and the unexpected complexity of the issue. Multiple iterations, which began in June, gradually improved the situation, but the absence of an airport workshop made testing new concepts time-consuming.

In January 2021, significant changes occurred in the regulations governing UAV flight permits in Germany. The authority responsible for issuing these permits shifted from the Bavarian authority to the National Aviation Authority of Germany (LBA). This transition led to longer processing times for permit applications. More importantly, the rules related to UAV flight safety, which were initially applied to the FLEXOP project in 2018, underwent substantial revisions. Under the new regulations, flight permits were only granted if applicants followed the risk assessment process known as SORA, provided by the European Union Aviation Safety Agency (EASA). These new rules were not only more detailed but also more stringent than the previous ones. The plan was to conduct a flight test campaign at the Cochstedt UAV Flight Test Centre, which is part of the German Aerospace Center (DLR), in late September or early October 2021. The application process for this permit began in mid-August. At that time, the LBA indicated that processing could take approximately six weeks. However, the team soon realized the extensive work required to submit a compliant application. This included recalculating flight areas, conducting additional simulations (particularly for parachute systems), updating manuals, and rewriting the risk analysis. Eventually, a comprehensive 100-page application was submitted on September 30, 2021, following multiple rounds of discussions and feedback with the LBA resulting in also postphoning

flight test phase 2. Additionally, the location of Cochstedt, being distant from Munich where the flight crew was based, posed logistical challenges in organizing the flight tests. Obtaining a permit for Oberpfaffenhofen was not feasible within the scope of the project, as the conditions for flight operations in more populated areas were even more complex. The -3-wing design faced delays due to the development of the multi-disciplinary design loop. A study was conducted to explore the possibility of retrofitting an existing -0- or -2-wing with the required hardware for -3-wing functionality, specifically attaching and actuating eight flaps per wing for shape control and load alleviation. This retrofit was deemed feasible and implemented.

An accident during the second flight test campaign in August 2022 necessitated the rebuilding of the fuselage. This incident led to a reevaluation of the project's flight test priorities, with a consensus to prioritize wing-1 flight tests, particularly active flutter control experiments, after the demonstrator's reconstruction. The focus shifted away from advanced wing manufacturing and integration.

To prevent similar accidents and enhance operational ease, various design changes were implemented. These changes, including subsystem improvements and cabling layout modifications, positively impacted usability. The introduction of a programmed interface for pilots and switches in the payload area streamlined turn-on and turn-off processes. Extending the fuel tank allowed for longer flights, lasting 30-40 minutes, optimizing each flight opportunity. Despite the time and effort invested in rebuilding the aircraft, these system improvements contributed to the successful completion of planned active flutter control experiments during the final -1 campaigns. Overall, the P-FLEX, now referred to as the new aircraft, has proven to be a reliable and efficient platform for safe operations.

Since the outcomes of WP3 were received towards the project's conclusion, there was insufficient time to conduct a comprehensive evaluation. Nevertheless, all partners express strong interest in carrying forward this work beyond the project's completion. The intensified focus on WP3 and its late-stage completion left limited time for WP4 towards the project's end.

6.1.4 WP4

The workpackage was scheduled to start later in the project and as problems accumulated regarding the overall project schedule the effort from scale-up was slightly sifted to flight testing. This meant that the originally envisioned parameter sweeps with different aspect ratio wings and the automatic iterative MDO optimisation with closed loop loads were not executed. Only manual data transfer and one aspect ratio was investigated from drag reduction and range increase perspectives showing 7% drag reduction potential on the baseline D150.

6.1.5 WP5

The main deviations from the original DOW regarding WP5 were already discussed in the accepted amendment. The collaborative work, exploitation and dissemination activities were all heavily impacted by COVID. Procurement plans were also heavily impacted by the demonstrator rebuild and the chip shortage crisis.

6.2 Sex and Gender Equality

The Consortium has identified that there are no special gender issues associated with the project subject. The consortium was always aware of the inherent participation inequality in engineering disciplines of sex and gender. The consortium members in all of their action, including recruiting and hiring practices acknowledged the principle of equality between women and men not only to eliminate inequalities, but also to promote equality, as signed in the Treaty on European Union, the Treaty of Amsterdam (May 1, 1999).

To better promote the project towards women, underrepresented within the project (3 female and 59 male participants), the team reached out to Bsc and Msc level University courses and promoted the project and encouraged students to join the team, where more female students without already selected topic are available.

The flight test team composition was specifically selected to include women on the team in most of the test campaigns, providing diversity.

The project manager Virág Bodor on the coordinator side conciosly took larger role in team building activities and consortium meetings to provide larger exposure to female opinions and a different perspective.

Despite all of our efforts the team was able to recruit only 2 female researchers, which is not unheard in a discipline dominated mostly by male researchers.

6.3 Use of resources

6.3.1 SZTAKI

Personel cost of SZTAKI was 18% higher than planned, due to more effort spent on rebuilding the demonstrator and increased attendance at flight testing and data analysis. A direct consequence of this is an increase of 38% travel cost versus planned. Equipment procurement was 17% lower than expected, since the team abandoned the -3 wing instrumentation and testing activities at the point of crashing the demonstrator. More importantly other goods and services are at 26% of the originally planned, since wing FBG sensors and interrogators were not purchased (based on a Consortium decision focus was shifted to flutter wings) and due to Covid the advisory group had very limited travel opportunities.

The claimed unforeseen cost (equal to 2 061.18 €) was associated with the replacement of the air data probe sensor of the demonstrator. A replacement air data probe was necessary as the previous backup got damaged in a minor accident, due to an aborted take-off of the aircraft, the air data probe and the fuselage of the aircraft got damaged. The main air data probe was destroyed in the accident when the complete demonstrator was lost. Since the schedule was very tight for the rebuild and the flight test campaign, the team decided to have key components available at the airfield as replacement parts.

One of the main project dissemination activity was the AIAA Scitech ’23 conference in Washington DC, USA, January 22-28. The total cost of the trip, including airfare, hotel, conference registration and per diem was 6315 €. The FliPASED consortium organized and hosted two invited sessions, AIAA 2023-0175 Session: Special Session: Design, Modeling and Testing of ASE Demonstrator for the FLEXOP and FLiPASED EU Project I and AIAA 2023-0372 Session: Special Session: Design, Modeling and Testing of ASE Demonstrator for the FLEXOP and FLiPASED EU Project II. The first session was co-hosted by NASA the second by Lockheed-Martin and both of them provided excellent opportunity to disseminate the project result not only to the scientific community but also to the industry experts. Within the conference the team also made an ad-hoc consortium meeting and briefed the industry advisory group member from Airbus about the project progress. The team also exchanged ideas and lessons-learnt with the NASA X-56 program.

5th Annual Learning for Dynamics & Control Conference (5th L4DC) was held at the University of Pennsylvania June 14-16, 2023. One researcher was participating, with a total cost of 2850 €. One of the novelty within the project was to combine traditional aeroleastic modelling tools with artificial intelligence and machine learning methods to demonstrate the applicability of big data methods associated with aerospace flight testing. Earlier results of the same research were shared with the audience at the IFASD 22 conference in Madrid with the aeroelasticity community, but since the results are multidisci-

plinary the most suitable conference to disseminate the results with the machine learning community was carefully selected to be the 5th L4DC. The project officer was notified in advance of these oversees conferences.

Travel costs: For the travels mentioned in the Report On Explanations On The Use Of Resources please see the detailed tables and brief explanations:

2022-02-20	2022-02-25	Gyulai László, Németo, München
2022-02-20	2022-02-25	Balogh Dániel, Németo, München
2022-02-21	2022-02-24	Tóth Szabolcs, Németo, München
2022-02-21	2022-02-24	Nagy Mihály, Németo, München
2022-02-21	2022-02-24	Fazekas Bálint, Németo, München
2022-02-21	2022-02-24	Fritsch Balázs, Németo, München
2022-02-21	2022-02-24	Novozánszki Zsombor, Németo, München

20.02.2022 – 25.02.2022. Travel of SZTAKI team (7 persons) to Münich, Germany. Development of demonstrator at TUM premises together with TUM team (integration of new flight control computer and flatter actuator) - 4600,50 EUR.

2022-08-19	2022-08-25	Fritsch Balázs, Németo, Cochstedt
2022-08-19	2022-08-25	Tóth Szabolcs, Németo, Cochstedt
2022-08-19	2022-08-25	Szabó Péter, Németo, Cochstedt
2022-08-19	2022-08-25	Balogh Dániel, Németo, Cochstedt

19.08.2022 – 26.08.2022. Travel of SZTAKI team (4 persons) to Cochstedt, Germany to support the Flight Test performed by FLIPASED team. Flight test was aiming to test the demonstrator's hardware, software and autopilot functionalities -5 078,25 EUR.

2023-01-22	2023-01-27	Takarics Béla, USA, National Harbour
2023-01-22	2023-01-28	Vanek Bálint, USA, National Harbour

22.01.2023. – 28.01.2023. One of the main project dissemination activity was the AIAA Scitech '23 conference in Washington DC, USA, January 22-28. Two researchers including coordinator have participated (Bálint VAnek and Béla Takarics). The total cost of the trip, including airfare, hotel, conference registration and per diem was 6315 €. The FliPASED consortium organized and hosted two invited sessions, AIAA 2023-0175 - 6 315,22 EUR.

2023-04-02	2023-04-06	Nagy Mihály, Németo, Cochstedt
2023-04-02	2023-04-06	Vanek Bálint, Németo, Cochstedt
2023-03-28	2023-04-01	Luspay Tamás, Németo, Cochstedt
2023-03-26	2023-04-06	Balogh Dániel, Németo, Cochstedt
2023-03-26	2023-04-01	Tóth Szabolcs, Németo, Cochstedt

26.03.2023. – 06.04.2023. Travel of SZTAKI team (5 persons) to Cochstedt, Germany to support the Flight Test performed by FLIPASED team. Testing of the rebuilt demonstrator and the flutter actuator - 7 599,76 EUR.

2023-05-08	2023-05-12	Nagy Mihály, Németo, Cochstedt
2023-05-07	2023-05-12	Balogh Dániel, Németo, Cochstedt
2023-05-07	2023-05-12	Vanek Bálint, Németo, Cochstedt

07.05.2023. – 12.05.2023. Travel of SZTAKI team (3 persons) to Cochstedt, Germany to support the Flight Test performed by FLIPASED team. Testing of the rebuilt demonstrator and the flutter actuator. Debugging of the telemetry system - 4 055,68 EUR.

2023-05-21	2023-05-26	Balogh Dániel, Németo., Cochstedt
2023-05-21	2023-05-26	Vanek Bálint, Németo, Cochstedt
2023-05-21	2023-05-26	Tóth Szabolcs, Németo, Cochstedt

21.05.2023. – 26.05.2023. Travel of SZTAKI team (3 persons) to Cochstedt, Germany to support the Flight Test performed by FLIPASED team. Perform the active flutter control tests - 4 900,51 EUR.

Please note that there are some changes in the explanation on the use of resources (explaining the exceeding amount of other direct costs). Sztaki has revisited the travel costs in this last iteration and we have regrouped some costs. Please note that there were quite complicated travels: when building and rebuilding the aircraft (Münich) and also at flight tests (Cochstedt) the support of SZTAKI's team was crucial. Different team members were needed in different stages of the process but what we considered as one work process we grouped it into one travel.

The WP related spending was aligned with the plan, only extra work on tools and methods were charged with an additional 34% on WP2, due to extra effort required from the Big Data analytics group to train neural networks for wingshape estimation.

CFS: 1/MTA SZTAKI:

- The period covered by the report/duration of the action is corrected in the ToR and we confirm that the reported costs were incurred between 01/09/2019 & 30/06/2023.
- The total costs declared in the independent audit report (EUR 1,154,631.81) did not correspond to those declared in the financial statements of the beneficiary (EUR 1,155,759.44). It is now corrected. The cause of this difference (1,127.62 euro) was that in the independent audit there were no indirect costs calculated on the cost of the audit itself (4,510.46 *0,25).
- Finding 36) is N/A in the procedures and now it is also listed in the Not applicable Findings section of the Independent Report.
- Findings 47-52 (D.2 Depreciation costs for equipment, infrastructure or other assets) is also corrected in the CFS.

The discrepancy between the amount indicated in the system and in the UoR report as CFS fee: i.e. EUR 4,510.46 and the amount indicated in the CFS: EUR 5,728.28 is the VAT. Now it is clearly stated in the CFS as well that the net amount of the audit cost is 4,510.46 euro while the gross amount is 5,728.28 euro. SZTAKI hereby confirms that no deductible VAT is included in the costs claimed in the financial report

"Equipment costs - various items claimed:" We confirm that the equipment costs claimed in RP2 refer indeed to purchase of assets/equipment and they have been calculated as depreciation for the portion corresponding to the reporting period and for the rate of actual use for the action (Art. 6.2).

6.3.2 DLR

More meetings and workshops with more staff attending than originally envisaged were happening, as soon as it was possible after the pandemic. In particular the "collaborative task definition" (Task 1.3) required intensive discussions and exchange during in person meetings and workshops, as well as many remote meetings to setup the workflow definition.

Equipment related costs were not declared on the project even though the hardware-in-the-loop platform was originally planned to be purchased from FliPASED funds. DLR used other project funds to purchase the equipment and did not want to duplicate them.

SZTAKI	Costs incurred in the 2nd period (M21-M46)	Costs incurred during the whole project (M1-M46)	Costs estimated for the whole project (46M)- based on amendment data	%
Personnel costs	534 111,89	757 368,65	644 277,70	118%
Subcontracting		0,00		
Other direct costs	149 930,18	167 238,90	231 500,00	72%
travel	82 814,73	86 217,55	62 500,00	138%
equipment	45 485,88	54 212,38	65 000,00	83%
other goods & services	21 629,57	26 808,97	104 000,00	26%
Indirect costs	171 010,52	231 151,89	218 944,43	106%
Total	855 052,59	1 155 759,49	1 094 722,13	106%
SZTAKI	Staff effort per WP in the 2nd period	Staff effort during the whole project (M1-M46)	Staff effort planned for the whole project (46M)	%
WP1	29,49	52,08	52,16	100%
WP2	47,47	70,14	52,20	134%
WP3	54,91	76,41	72,61	105%
WP4	24,68	24,68	24,91	99%
WP5	29,99	50,67	50,44	100%
total	186,54	273,98	252,32	109%

Figure 460: Use of resources for SZTAKI

Adjustments: 3/DLR: ”Please enter in the UoR and briefly explain the Other Direct Cost (ODC) adjustment.” An adjustment was made to RP1 of DLR financial statement. It was made because a correction was necessary for the deduction of VAT within the travel expenses. The adjustment concerned the item hotel for Flipased Kick-off Meeting in September 2019 and the items hotel, public transport, taxi for Project Meeting in November 2019.

In the Financial Statement for RP1, the correction to the VAT for foreign travel costs was too high. Therefore the adjustment in RP2 adds back the deduction. This was corrected as part of the Financial Statement for RP2.

This is the reason the adjustment amount is positive.

DLR	Costs incurred in the 2nd period (M21-M46)	Costs incurred during the whole project (M1-M46)	Costs estimated for the whole project (46M)-based on amendment data	%
Personnel costs	471 637,2	731 889,35	697 473,10	104,93%
Subcontracting	0,0	0,00	0,00	
Other direct costs	33 144,1	39 585,43	78 000,00	50,75%
travel	23 603,6	25 946,83	32 000,00	81,08%
equipment	0,0	0,00	30 000,00	0,00%
other goods & services	9 540,5	13 638,60	16 000,00	85,24%
Indirect costs	126 195,3	192 868,70	193 868,28	99,48%
Total	630 976,6	964 343,48	969 341,38	99,48%
DLR	Staff effort per WP (hours) in the 2nd period	Staff effort during the whole project (M1-M46)	Staff effort planned for the whole project (46M)	%
WP1	9,78	16,44	12,00	137,00%
WP2	3,94	25,17	27,50	91,53%
WP3	22,29	31,21	29,00	107,62%
WP4	30,21	31,08	29,00	107,17%
WP5	0,00	0,00	0,00	
total	66,22	103,90	97,50	106,56%

Figure 461: Use of resources for DLR

6.3.3 ONERA

The original contribution of ONERA was reduced during the amendment, since key personell was already booked for the extension period (last 6 months of the project). On the other hand the reduced budget was overspent by 36% since increased effort was required during the GVT (happening in the project extension period) and also flight testing was supported by ONERA, orignally not planned, since the flutter predictions of the demonstrator had to be updated in the last period of the project. The increased travel expenses are a direct consequence of these extra efforts. WP1 is overspent by 71% which is related to the overall setup of the demonstrator and the increased need of collaboration, while WP3 is severely overspent (1490%) since GVT and flight testing activities were originally planned to be conducted by DLR and ONERA took over this task.

ONERA	Costs incurred in the 2nd period (M21-M46)	Costs incurred during the whole project (M1-M46)	Costs estimated for the whole project (46M)- based on amendment data	%
Personnel costs	236 490,27	330 237,00	243 227,00	136%
Subcontracting	0,00	0,00	0,00	
Other direct costs	10 469,61	13 679,28	7 841,95	174%
travel	9 922,37	13 132,04	7 841,95	167%
equipment	0,00	0,00	0,00	
other goods & services	547,24	547,24	0,00	
Indirect costs	61 739,97	85 979,07	62 767,24	137%
Total	308 699,85	429 895,35	313 836,19	137%
		429 895,35		
ONERA	Staff effort per WP (hours) in the 2nd period	Staff effort during the whole project (M1-M46)	Staff effort planned for the whole project (46M)	%
WP1	1,49	5,14	3,00	171%
WP2	5,52	13,57	15,00	90%
WP3	27,69	29,79	2,00	1490%
WP4	0,00	0,00	15,00	0%
WP5	0,00	0,00	0,00	
total	34,70	48,50	35,00	139%

Figure 462: Use of resources for ONERA

6.3.4 TUM

Tum spent slightly more man hours in WP3 as initially planned due to increased effort due to the change in UAV regulations and the necessary application and certification process, also due to the change of the flight test location to Cochstedt, which is farther away from Munich and required to plan flight test campaigns rather than single test days. Also the repair of the aircraft took unplanned resources,

especially at the end of the project. Therefore less manpower was available for WP4 which was also scheduled mainly at the end of the project. But in an overall perspective over all work packages TUM did spend almost exactly the planned man hours.

TUM	Costs incurred in the 2nd period (M21-M46)	Costs incurred during the whole project (M1-M46)	Costs estimated for the whole project (46M)-based on amendment data	%
Personnel costs	709 833,08	1 097 016,49	1 073 585,41	102,18%
Subcontracting		0,00		
Other direct costs	112 711,26	126 721,96	101 090,22	125,36%
travel	50 809,56	51 698,26	41 090,20	125,82%
equipment		13 122,00	2 000,00	656,10%
other goods & services	61 901,70	61 901,70	58 000,00	106,73%
Indirect costs	205 636,09	305 934,62	293 668,91	104,18%
Total	1 028 180,43	1 529 673,07	1 468 344,54	104,18%
TUM	Staff effort per WP (hours) in the 2nd period	Staff effort during the whole project (M1-M46)	Staff effort planned for the whole project (46M)	%
WP1	0,98	24,99	26,00	96,12%
WP2	18,34	27,95	26,00	107,50%
WP3	102,12	141,81	138,00	102,76%
WP4	4,37	4,37	12,00	36,42%
WP5				
total	125,81	199,12	202,00	98,57%

Figure 463: Use of resources for TUM

Financial deviations: Even though travel was initially limited due to COVID-19 at the project's outset, the travel expenses ended up exceeding the planned budget. The primary factor behind this was the essential decision to relocate flight tests from Oberpfaffenhofen to Cochstedt. Oberpfaffenhofen is in close proximity to Munich, allowing for one-day flight tests without the need for overnight stays. However, the new location in Cochstedt necessitated multi-day or week-long flight test campaigns, requiring accommodation. Additionally, post-COVID, the travel cost, particularly for flights, became more expensive. In the original project proposal, the equipment costs were initially estimated at 50.000 Euro. In the subsequent amendment, TUM restructured the budget, allocating the majority of these costs to personnel expenses, retaining only 2.000 Euro. This adjustment was made in recognition of the increased effort required in WP3. However, despite these intended cost-saving measures, we ultimately spent 13.122 Euro to rebuild the demonstrator and procure all the necessary measurement equipment essential for completing the flutter flight tests.

7 Bibliography

[1] *MATLAB - MathWorks - MATLAB & Simulink*.

[2] A.C. Antoulas. *Approximation of large-scale dynamical systems*. SIAM, Philadelphia, 2005.

[3] A.C. Antoulas, C.A. Beattie, and S. Gugercin. *Interpolatory methods for model reduction*. SIAM Computational Science and Engineering, Philadelphia, 2020.

[4] Gary J Balas, Andrew Packard, Peter J Seiler, and Arnar Hjartarson. *LPVTools - A toolbox for modeling, analysis and synthesis of parameter yarying control systems*. MUSYN Inc., 2015.

[5] Julius Bartasevicius and Mirko Hornung. In-flight drag measurement and validation for a medium-sized UAV. In *AIAA SCITECH 2023 Forum*, number January, pages 1–13, Reston, Virginia, jan 2023. American Institute of Aeronautics and Astronautics.

[6] Julius Bartasevicius, Sebastian J Koeberle, Daniel Teubl, Christian Roessler, and Mirko Hornung. Flight testing of 65kg t-flex subscale demonstrator. In *32nd Congress of the International Council of the Aeronautical Sciences*, pages 1–16. ICAS, 2021.

[7] G. Becker. *Quadratic stability and performance of linear parameter dependent systems*. PhD thesis, University of California, Berkeley, 1993.

[8] Alberto Bemporad, N. Lawrence Ricker, and Manfred Morari. Model Predictive Control Toolbox User's Guide. 2022.

[9] Raymond L. Bisplinghoff, Holt Ashley, and Robert L. Halfman. *Aeroelasticity*. Dover Publications, Inc., 1955.

[10] Brigitte Boden, Jan Flink, Niklas Först, Robert Mischke, Kathrin Schaffert, Alexander Weinert, Annika Wohlan, and Andreas Schreiber. RCE: An integration environment for engineering and science. *SoftwareX*, 15:100759, July 2021.

[11] Brigitte Boden, Jan Flink, Robert Mischke, Kathrin Schaffert, Alexander Weinert, Annika Wohlan, Caslav Ilic, Tobias Wunderlich, Carsten M. Liersch, Stefan Görtz, Erwin Moerland, and Pier Davide Ciampa. Distributed Multidisciplinary Optimization and Collaborative Process Development Using RCE. In *AIAA Aviation 2019 Forum, 17–21 June 2019, Dallas, TX, USA*. American Institute of Aeronautics and Astronautics, 2019.

[12] David A. Burdette and Joaquim R.R.A. Martins. Design of a transonic wing with an adaptive morphing trailing edge via aerostructural optimization. *Aerospace Science and Technology*, 81:192–203, 2018.

[13] A. W. Burner, W. A. Lokos, and D. A. Barrows. In-flight aeroelastic measurement technique development. In *Proceedings of SPIE - The International Society for Optical Engineering*, pages 1–14, 2003.

[14] I.R. Chittick and J.R.R.A. Martins. An asymmetric suboptimization approach to aerostructural optimization. *Optimization and Engineering*, 10:133–152, 2009.

[15] Systems Concepts. Flight Test Measurement Techniques for Laminar Flow (Les techniques de mesure en vol. Technical Report October, 2003.

[16] NVIDIA Corporation. GEFORCE RTX 3060 FAMILY, 2021. Available from: `https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3060-3060ti/` [Accessed: 15/08/2021].

[17] A. Da Ronch, C. McFarlane, C. Beaverstock, J. Oppelstrup, M. Zhang, and A. Rizzi. Benchmarking ceasiom software to predict flight control and flying qualities of the B-747. *27th Congress of the International Council of the Aeronautical Sciences 2010, ICAS 2010*, 4(September):2906–2912, 2010.

[18] Andre Deperrois. Xflr5.

[19] André Deperrois. Guidelines for XFLR5: Analysis of foils and wings operating at low Reynolds numbers. Technical Report February, 2013.

[20] Andre Deperrois. Xflr5, 29.05.2022.

[21] Mark Drela. Avl.

[22] Mark Drela. Xfoil.

[23] Quentin Arnaud Dugne-Hennequin, Hideaki Uchiyama, and João Paulo Silva Do Monte Lima. Understanding the behavior of data-driven inertial odometry with kinematics-mimicking deep neural network. *IEEE Access*, 9:36589–36619, 2021.

[24] European Aviation Safety Agency. Certification specifications for large aeroplanes (cs-25). Technical report, European Aviation Safety Agency, 2007.

[25] O. Faugeras. Three-dimensional computer vision: A geometric viewpoint. In *Artificial Intelligence Series*. MIT Press, 1993.

[26] Jan Flink, Robert Mischke, Kathrin Schaffert, Dominik Schneider, and Alexander Weinert. Orchestrating tool chains for model-based systems engineering with RCE. In *2022 IEEE Aerospace Conference (AERO)*. IEEE, March 2022.

[27] Edward H. Frank and Jim Lyle. *Type-Certificate Data Sheet No. EASA.A.064 for Airbus A318-A319-A320-A321*. European Aviation Safety Agency, 2012.

[28] Alessandro Gastaldi and Aaron Dettmann. Pytornado.

[29] I.V. Gosea, S. Gugercin, and C. Beattie. Data-driven balancing of linear dynamical systems. *arXiv preprint arXiv:2104.01006*, 2021.

[30] Tomasz Grabowski. Panukl, 2022.

[31] S. Görtz, C. Ilic, M. Abu-Zurayk, R. Liepelt, J. Jepsen, T. Führer, R. Becker, J. Scherer, T. Kier, and M. Siggel. Collaborative multi-level mdo process development and application to long-range transport aircraft. In *30th International Congress of the Aeronautical Sciences, Daejeon, South Korea, September 25-30. 2016*. ICAS, 2016.

[32] Stefan Görtz, Mohammad Abu-Zurayk, Caslav Ilic, Tobias F. Wunderlich, Stefan Keye, Matthias Schulze, Christoph Kaiser, Thomas Klimmek, Özge Süelözgen, Thiemo Kier, Andreas Schuster, Sascha Dähne, Michael Petsch, Dieter Kohlgrüber, Jannik Häßy, Robert Mischke, Alexander Weinert, Philipp Knechtges, Sebastian Gottfried, Johannes Hartmann, and Benjamin Fröhler. Overview of Collaborative Multi-Fidelity Multidisciplinary Design Optimization Activities in the DLR Project VicToria. In *AIAA AVIATION 2020 FORUM*. American Institute of Aeronautics and Astronautics, jun 2020.

[33] Vega Handojo. *Contribution to Load Alleviation in Aircraft Pre-design and Its Influence on Structural Mass and Fatigue*. PhD thesis, Deutsches Zentrum fuur Luft- und Raumfahrt Institut fuur Aeroelastik, Gottingen, 2020.

[34] Vladimir Hanta and Aleš Procházka. Rational approximation of time delay. *Journal*, 2009.

[35] S. Hedman. Vortex Lattice Method for Calculation of Quasi Steady State Loadings on Thin Elastic Wings. Technical Report Report 105, Aeronautical Research Institute of Sweden, October 1965.

[36] J. Hofstee, T. Kier, C. Cerulli, and G. Looye. A Variable, Fully Flexible Dynamic Response Tool for Special Investigations (VarLoads). In *International Forum on Aeroelasticity and Structural Dynamics*, 2003.

[37] 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.

[38] Caslav Ilic, Andrei Merle, Mohammad Abu-Zurayk, Stefan Görtz, Martin Leitner, Özge Süelözgen, Thiemo Kier, Matthias Schulze, Thomas Klimmek, Christoph Kaiser, David Quero, Andreas Schuster, Michael Petsch, Dieter Kohlgrüber, Jannik Häßy, Richard Becker, and Sebastian Gottfried. Cybermatrix Protocol: A Novel Approach to Highly Collaborative and Computationally Intensive Multidisciplinary Aircraft Optimization. In *AIAA AVIATION 2020 FORUM*. American Institute of Aeronautics and Astronautics, jun 2020.

[39] Insta360. Insta 360 ONE X, 2021. Available from: `https://www.insta360.com/es/product/insta360-onex/` [Accessed: 30/08/2021].

[40] Intel. Intel® Core™ i5-10600K Processor, 2020. Available from: `https://www.intel.com/content/www/us/en/products/sku/199311/intel-core-i510600k-processor-12m-cache-up-to-4-80-ghz/specifications.html` [Accessed: 15/08/2021].

[41] J-F. Magni. Linear Fractional Representation toolbox for use with Matlab. Available with the SMAC Toolbox at: `http://w3.onera.fr/smac/lfrt`, downloaded on 26th June 2023.

[42] R. Jategaonkar and R.V. Jategaonkar. *Flight Vehicle System Identification: A Time Domain Methodology*. Progress in astronautics and aeronautics. American Institute of Aeronautics and Astronautics, 2006.

[43] R. Jategaonkar and R.V. Jategaonkar. *Flight Vehicle System Identification: A Time Domain Methodology*. Progress in astronautics and aeronautics. American Institute of Aeronautics and Astronautics, 2006.

[44] M. Karpel, B. Moulin, and P. C. Chen. Dynamic response of aeroservoelastic systems to gust excitation. *Journal of Aircraft*, 42(5):1264–1272, 2005.

[45] Joseph Katz and Allen Plotkin. *Low-Speed Aerodynamics*. Cambridge University Press, New York, 2nd edition, 2001.

[46] Gaetan K. W. Kenway and Joaquim R. R. A. Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51(1):144–160, jan 2014.

[47] T. M. Kier and J. Hofstee. VarLoads - eine Simulationsumgebung zur Lastenberechnung eines voll flexible, freifliegenden Flugzeugs. In *Deutscher Luft- und Raumfahrtkongress 2004 20.-23. September 2004, Dresden*. DGLR, 2004.

[48] Thiemo Kier. An Integral 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, 13-17 June 2022, Madrid, Spain*, number IFASD-2022-093, 2022.

[49] 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.

[50] Thiemo Kier and Gertjan Looye. Unifying Manoeuvre and Gust Loads Analysis Models. In *International Forum on Aeroelasticity and Structural Dynamics*, 2009.

[51] David Kinney. Using VSPAERO.

[52] David Kinney. VSPAERO... What's New?, 2021.

[53] Thomas Klimmek. Parameterization of Topology and Geometry for the Multidisciplinary Optimization of Wing Structures. page 9, 2009.

[54] Thomas Klimmek. *Statische aeroelastische Anforderungen beim multidisziplinären Strukturentwurf von Transportflugzeugflügeln*. PhD thesis, DLR - Institut für Aeroelastik, 2016.

[55] Thomas Klimmek, Thiemo Kier, Andreas Schuster, Tobias Bach, Dieter Kohlgrüber, Matthias Schulze, and Martin Leitner. Loads Analysis and Structural Optimization - A Parameterized and Integrated Process. In *1st European Workshop on MDO for Industrial Applications in Aeronautics - Challenges and Expectations, 24.-25. Okt. 2017, Braunschweig*, 2017.

[56] 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.

[57] Sebastian J. Koeberle, Julius Bartasevicius, Daniel Teubl, and Szabolcs Tóth. Flipased D3.5 wing -3 manufacturing design for advanced wing. Technical report, TUM, 2022.

[58] Raymond M. Kolonay and Franklin E. Eastep. Optimal scheduling of control surfaces flexible wings to reduce induced drag. *Journal of Aircraft*, 43(6):1655–1661, 2006.

[59] Aditya Kotikalpudi, Brian P Danowsky, David K Schmidt, Christopher D Regan, and Abhineet Gupta. Real-time shape estimation for a small flexible flying-wing aircraft. In *AIAA Scitech 2019 Forum*, page 1818, 2019.

[60] Wolf R. Krüger, Yasser M. Meddaikar, Johannes K. S. Dillinger, Jurij Sodja, and Roeland De Breuker. Application of Aeroelastic Tailoring for Load Alleviation on a Flying Demonstrator Wing. *Aerospace*, 9(10), 2022.

[61] F. Leibfritz and W. Lipinski. Description of the benchmark examples in $COMP_{e}ib$ 1.0. Technical report, University of Trier, 2003.

[62] George Loubimov and Michael Kinzel. A novel approach to calculating induced drag from computational fluid dynamics. *AIP Advances*, 11(7), jul 2021.

[63] Tamás Luspay, Daniel Ossmann, Matthias Wuestenhagen, Dániel Teubl, Tamás Baár, Manuel Pusch, Thiemo M. Kier, Sergio Waitman, Andrea Ianelli, Andres Marcos, Balint Vanek, and Mark H. Lowenberg. Flight control design for a highly flexible flutter demonstrator. In *AIAA Scitech 2019 Forum*. AIAA, jan 2019.

[64] Leandro R Lustosa, Ilya Kolmanovsky, Carlos ES Cesnik, and Fabio Vetrano. Aided inertial estimation of wing shape. *Journal of Guidance, Control, and Dynamics*, 44(2):210–219, 2021.

[65] RE Maine and KW Iliff. Agard flight test techniques series. volume 3. identification of dynamic systems-applications to aircraft. part 1. the output error approach. Technical report, ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT NEUILLY-SUR-SEINE (FRANCE), 1986.

[66] Manuel Pusch. *Blending of Inputs and Outputs for Modal Control of Aeroelastic Systems: Dissertation*. Verlag Dr. Hut, München, 2020.

[67] Andres Marcos, Subhabrata Ganguli, and Gary J Balas. An application of H_∞ fault detection and isolation to a transport aircraft. *Control Engineering Practice*, 13(1):105–119, 2005.

[68] J.R.R.A. Martins. Multidisciplinary design optimization of aerospace systems. *Advances and Trends in Optimization with Engineering Applications*, pages 249–257, 2017.

[69] Brian Maskew. Program VSAERO theory document. *Nasa Cr-4023*, 1987.

[70] Yasser M. Meddaikar, Johannes Dillinger, Thomas Klimmek, Wolf Krueger, Matthias Wuestenhagen, Thiemo M. Kier, Andreas Hermanutz, Mirko Hornung, Vladyslav Rozov, Christian Breitsamter, James Alderman, Bela Takarics, and Balint Vanek. Aircraft aeroservoelastic modelling of the FLEXOP unmanned flying demonstrator. In *AIAA Scitech 2019 Forum*. AIAA, jan 2019.

[71] Yasser M. Meddaikar, Johannes Dillinger, Thomas Klimmek, Wolf Krueger, Matthias Wuestenhagen, Thiemo M. Kier, Andreas Hermanutz, Mirko Hornung, Vladyslav Rozov, Christian Breitsamter, James Alderman, Bela Takarics, and Balint Vanek. Aircraft aeroservoelastic modelling of the FLEXOP unmanned flying demonstrator. In *AIAA Scitech 2019 Forum*. AIAA, jan 2019.

[72] Yasser M. Meddaikar, Johannes Dillinger, Jurij Sodja, and Roeland De Breuker. FLEXOP - application of aeroelastic tailoring to a flying demonstrator wing. In *DGLR*. DGLR, 2018.

[73] Tomas Melin. A vortex lattice matlab implementation for linear aerodynamic wing applications, 12 2000.

[74] Tomas Melin. Implementation of a vortex lattice method in a heterogeneous programming language environment, 2018.

[75] Microsoft. Windows 10 Pro, 2021. Available from: `https://www.microsoft.com/de-de/d/windows-10-pro/` [Accessed: 15/08/2021].

[76] Mobius. Instruction Manual for the Mobius ActionCam), 2015. Available from: `http://www.mobius-actioncam.com/wp-content/uploads/2015/01/Mobius-Manual-23jan15a.pdf` [Accessed: 15/06/2021].

[77] Mobius. Mobius 1080p HD Action Camera Standard Lens pack (Lens A2), 2021. Available from: `https://www.mobius-cam.com/en/mobius-1-action-camera-c-29_6/mobius-1080p-hd-action-camera-standaard-lens-set-lens-a2-p-48?zenid=967hkuga6fhav17qks70u92ls6` [Accessed: 15/06/2021].

[78] J. Moran. *An Introduction to Theoretical and Computational Aerodynamics*. Dover Books on Aeronautical Engineering. Dover Publications, 2003.

[79] Ábel Olgyay, Béla Takarics, Bence Körösparti, János Lelkes, Csaba Horváth, and Bálint Vanek. Aeroservoelasticity investigation with panel method. In *The 18th International Conference on Fluid Flow Technologies, Budapest - Hungary*, 2022.

[80] Erik D. Olson and Cindy W. Albertson. Aircraft high-lift aerodynamic analysis using a surface-vorticity solver. volume 0. American Institute of Aeronautics and Astronautics Inc, AIAA, 2016.

[81] OpenCV. Camera Calibration, 2021. Available from: https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html [Accessed: 18/08/2021].

[82] OpenCV team. About, 2020. Available from: https://opencv.org/about/ [Accessed: 03/05/2021].

[83] OpenVSP. Software package, version 3.22.0, 2021.

[84] Daniel Ossmann, Tamas Luspay, and Balint Vanek. Baseline flight control system design for an unmanned flutter demonstrator. In *2019 IEEE Aerospace Conference*, pages 1–10. IEEE, 2019.

[85] Bálint Patartics, György Lipták, Tamás Luspay, Peter Seiler, Béla Takarics, and Bálint Vanek. Application of structured robust synthesis for flexible aircraft flutter suppression. *IEEE Transactions on Control Systems Technology*, 30(1):311–325, 2021.

[86] Bálint Patartics, György Lipták, Tamás Luspay, Peter Seiler, Béla Takarics, and Bálint Vanek. Application of structured robust synthesis for flexible aircraft flutter suppression. *IEEE Transactions on Control Systems Technology*, pages 1–15, 2021.

[87] W F Phillips and D O Snyder. Modern Adaptation of Prandtl's Classic Lifting-Line Theory. *JOURNAL OF AIRCRAFT*, 37(4), 2000.

[88] C. Poussot-Vassal, P. Kergus, and P. Vuillemin. *Realization and Model Reduction of Dynamical Systems, A Festschrift in Honor of the 70th Birthday of Thanos Antoulas (eds. C. Beattie, P. Benner, M. Embree, S. Gugercin and S. Lefteriu)*, chapter Interpolation-based irrational model control design and stability analysis, pages 353–371. Springer, 2022.

[89] Charles Poussot-Vassal, Julius Bartasevicius, and Balint Vanek. Flipased D3.1 flight test programme – flight test phase #1. Technical report, SZTAKI, 2021.

[90] Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud JG Van Sloun, and Yonina C Eldar. Kalmannet: Neural network aided kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022.

[91] Christian Roessler, Philipp Stahl, Franz Sendner, Andreas Hermanutz, Sebastian Koeberle, Julius Bartasevicius, Vladyslav Rozov, Christian Breitsamter, Mirko Hornung, Yasser M. Meddaikar, Johannes Dillinger, Jurij Sodja, Roeland De Breuker, Christos Koimtzoglou, Dimitrios Kotinis, and Panagiotis Georgopoulos. Aircraft design and testing of FLEXOP unmanned flying demonstrator to test load alleviation and flutter suppression of high aspect ratio flexible wings. In *AIAA Scitech 2019 Forum*. AIAA, jan 2019.

[92] D. Scaramuzza, A. Martinelli, and R. Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, New York, NY, USA, January 2006. IEEE.

[93] Simon Schelle. *Investigation and Implementation of Airspeed Calibration Methods based on UAV Flight Test Data*. Master thesis, Technical University of Munich, 2022.

[94] 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.

[95] J. S. Shamma. *Analysis and design of gain scheduled control systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, 1988.

[96] Muhammad K Shereen, Muhammad I Khan, Naeem Khan, and Wasi Ullah. By the design and implementation of modified kalman filter for lpv systems. *International Journal of Engineering Works*, 3(4):26–31, 2016.

[97] Jianbo Shi and Carlo Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, USA, June 1994. IEEE.

[98] João Paulo Silva do Monte Lima, Hideaki Uchiyama, and Rin-ichiro Taniguchi. End-to-end learning framework for imu-based 6-dof odometry. *Sensors*, 19(17):3777, 2019.

[99] James C Sivells and Robert H Neelly. Method for calculating wing characteristics by lifting-line theory using nonlinear section lift data. *National Advisory Commitee for Aeronautics*, (865):75–93, 1947.

[100] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and design*. John Wiley and Sons, 2 edition, 2005.

[101] Jurij Sodja, Roeland De Breuker, Yasser M. Meddaikar, Johannes K. Dillinger, Keith Soal, Yves Govers, Wolf Krueger, Panagiotis Georgopoulos, Christos Koimtzoglou, Christian Roessler, Sebastian J. Koeberle, Julius Bartasevicius, Daniel Teubl, Laszlo Gyulai, Szabolcs Toth, Mihaly Nagy, Daniel Balogh, Miklos Jasdi, Péter Bauer, and Balint Vanek. *Ground Testing of the FLEXOP Demonstrator Aircraft*.

[102] H.-J. Steiner. Preliminary design tool for propeller-wing aerodynamics part ii: Theory, 2010.

[103] Özge Süelözgen and Matthias Wüstenhagen. Operational modal analysis for simulated flight flutter test of an unconventional aircraft. *IFASD, The International Forum on Aeroelasticity and Structural Dynamics, 9-13 June 2019, Savannah, Georgia, USA*, 2019.

[104] Özge Süelözgen. A novel updating algorithm for linearized state-space models of an unmanned flexible aircraft using flight test data. *AIAA SCITECH 2022 Forum*, 2022.

[105] Béla Takarics and Bálint Vanek. Tensor product model-based robust flutter control design for the FLEXOP aircraft. *IFAC-PapersOnLine*, 52(12):134–139, 2019. 21st IFAC Symposium on Automatic Control in Aerospace ACA 2019.

[106] Béla Takarics and Bálint Vanek. Robust control design for the flexop demonstrator aircraft via tensor product models. *Asian Journal of Control*, 23(3):1290–1300, 2021.

[107] Bela Takarics, Balint Vanek, Aditya Kotikalpudi, and Peter Seiler. Flight control oriented bottom-up nonlinear modeling of aeroelastic vehicles. In *2018 IEEE Aerospace Conference*. IEEE, mar 2018.

[108] Béla Takarics, Bálint Vanek, Aditya Kotikalpudi, and Peter Seiler. Flight control oriented bottom-up nonlinear modeling of aeroelastic vehicles. In *2018 IEEE aerospace conference*, pages 1–10. IEEE, 2018.

[109] Béla Takarics, Bálint Patartics, Tamás Luspay, Balint Vanek, Christian Roessler, Julius Bartasevicius, Sebastian J. Koeberle, Mirko Hornung, Daniel Teubl, Manuel Pusch, Matthias Wustenhagen, Thiemo M. Kier, Gertjan Looye, Péter Bauer, Yasser M. Meddaikar, Sergio Waitman, and Andres Marcos. *Active Flutter Mitigation Testing on the FLEXOP Demonstrator Aircraft*.

[110] TechSmith. Camtasia 2020, 2020. Available from: `https://www.techsmith.com/camtasia-2020-press-release.html` [Accessed: 15/06/2021].

[111] Daniel Teubl, Szabolcs Toth, and Balint Vanek. Flipased D3.7 manufacturing advanced wing and fuselage finalized. Technical report, TUM, 2023.

[112] H. Theil. *Economic Forecasts and Policy*. Contributions to economic analysis. North-Holland Publishing Company, 1975.

[113] Szabolcs Tóth, László Gyulai, Dániel Balogh, Mihály Nagy, Bálint Vanek, Daniel Teubl, Sebastian Köberle, Julius Bartasevicius, Christian Rößler, Yves Govers, Keith Soal, Yasser M. Meddaikar, Johannes Dillinger, Jurij Sodja, Panos Georgopoulos, and Christos Koimtzoglou. D4.2: Vehicle ground test report including hils results. Technical report, DLR, 2019.

[114] Balint Vanek. Flexop D4.10 release of the flight test results and models of the a/c for the community. Technical report, SZTAKI, 2019.

[115] J. C. Vassberg, M. A. DeHaan, S. M. Rivers, and R. A. Wahls. Development of a common research model for applied cfd validation studies. In *AIAA Paper 2008-6919, 26th AIAA Applied Aerodynamics Conference, Hawaii, HI*, 2008.

[116] G. Vinnicombe. *Measuring Robustness of Feedback Systems*. PhD thesis, Univ. Cambridge, Cambridge, 1993.

[117] Fen Wu. *Control of Linear Parameter Varying Systems*. PhD thesis, Univ. California, Berkeley, 1995.

[118] Matthias Wuestenhagen, Thiemo Kier, Yasser M. Meddaikar, Manuel Pusch, Daniel Ossmann, and Andreas Hermanutz. Aeroservoelastic modeling and analysis of a highly flexible flutter demonstrator. In *2018 Atmospheric Flight Mechanics Conference*. AIAA, jun 2018.

[119] Matthias Wüstenhagen, Thiemo Kier, Yasser M Meddaikar, Manuel Pusch, Daniel Ossmann, and Andreas Hermanutz. Aeroservoelastic modeling and analysis of a highly flexible flutter demonstrator. In *2018 atmospheric flight mechanics conference*, page 3150, 2018.

[120] M. Wüstenhagen, Ö. Süelözgen, L. Ackermann, and J. Bartaševicius. Validation and update of an aeroservoelastic model based on flight test data. *IEEE Aerospace Conference*, 2021.

[121] Fanglin Yu, Julius Bartasevicius, and Mirko Hornung. Comparing potential flow solvers for aerodynamic characteristics estimation of the t-flex uav. 2022.

[122] Fanglin Yu, Julius Bartasevicius, and Mirko Hornung. Comparing potential flow solvers for aerodynamic characteristics estimations of the t-flex uav. In *ICAS 2022 - 33rd Congress of the International Council of the Aeronautical Sciences*, 2022.

[123] Fanglin Yu, Daniel Teubl, Szabolcs Tóth, Nicolas Guérin, Keith Soal, Thiemo Kier, and Özge Süelözgen. Flipased D3.10 flight readiness review demonstrator with advanced wing. Technical report, TUM, 2023.

[124] Ming Zhang, Mingming Zhang, Yiming Chen, and Mingyang Li. Imu data processing for inertial aided navigation: A recurrent neural network based approach. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3992–3998. IEEE, 2021.

[125] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.

[126] K. Zhou and J C. Doyle. *Essentials Of Robust Control*. Prentice Hall, 1997.

[127] Thomas Zill, Pier Davide Ciampa, and Björn Nagel. Multidisciplinary design optimization in a collaborative distributed aircraft design system. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 553, 2012.

[128] Özge Süelözgen. A novel updating algorithm for linearized state-space models of an unmanned flexible aircraft using flight test data. In *AIAA SCITECH 2022 Forum*, 2022.

[129] Özge Süelözgen. Application and validation of a model updating approach for linearized state-space models of flexible aircrafts using multiple flight test data. In *AIAA SCITECH 2023 Forum*, 2023.

[130] Özge Süelözgen and Gertjan Looye. *Application and Validation of a New Updating Algorithm for Linearized State-Space Models of Flexible Aircrafts Using Flight Test Data, IFASD 2022-157*.

[131] Özge Süelözgen and Matthias Wuestenhagen. Operational modal analysis for simulated flight flutter test of an unconventional aircraft. In *International Forum on Aeroelasticity and Structural Dynamics, IFASD 2019*, 2019.