FliPASED
WWW.FLIPASED.EU

D1.3 Demonstrator Ground and Flight Test Requirements Definition

Keith Soal, Richard Kuchar (DLR), Julius Bartasevicius (TUM),

Charles Poussot-Vassal (ONERA), Szabolcs Tóth (SZTAKI)

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

Lead Beneficiary for this deliverable:	SZTAKI
Last modified: 04/02/2022	**Status:** Delivered
Due date: 30/11/2020	

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

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

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

Glossary

ASE	Aeroservoelastic
AFS	Active Flutter Suppression
CAD	Computer-aided Design
CPACS	Common Parametric Aircraft Configuration Schema
DLM	Doublet Lattice Method
FE	Finite Element
GLA	Gust Load Alleviation
LPV	Linear Parameter-varying
LPI	Linear Time-invariant
MDAx	MDAO Workflow Design Accelerator
MDO	Multidisciplinary Design Optimization
MIMO	Multi-Input Multi-Output
MLA	Manoeuvre Load Alleviation
PID	Proportional-Integral-Derivative
RCE	Remote Component Environment
ROM	Reduced Order Model
TCL	Tool Command Language
W3C	Wold Wide Web Consortium
XDSM	Extended Design Structure Matrix
XML	Extensible Markup Language
XSD	XML Schema Definition

Table of contents

List of Figures

1 Executive Summary

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.

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 ground and flight testing requirements and their relationship with the overall aims of the project.

2 Demonstrator Status

2.1 Demonstrator Airframe

The two goals of the legacy FLEXOP project, namely aeroelastic tailoring and active flutter suppression, presented very different requirements for flight tests [3]. 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 1. The demonstrator is mostly flown 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.

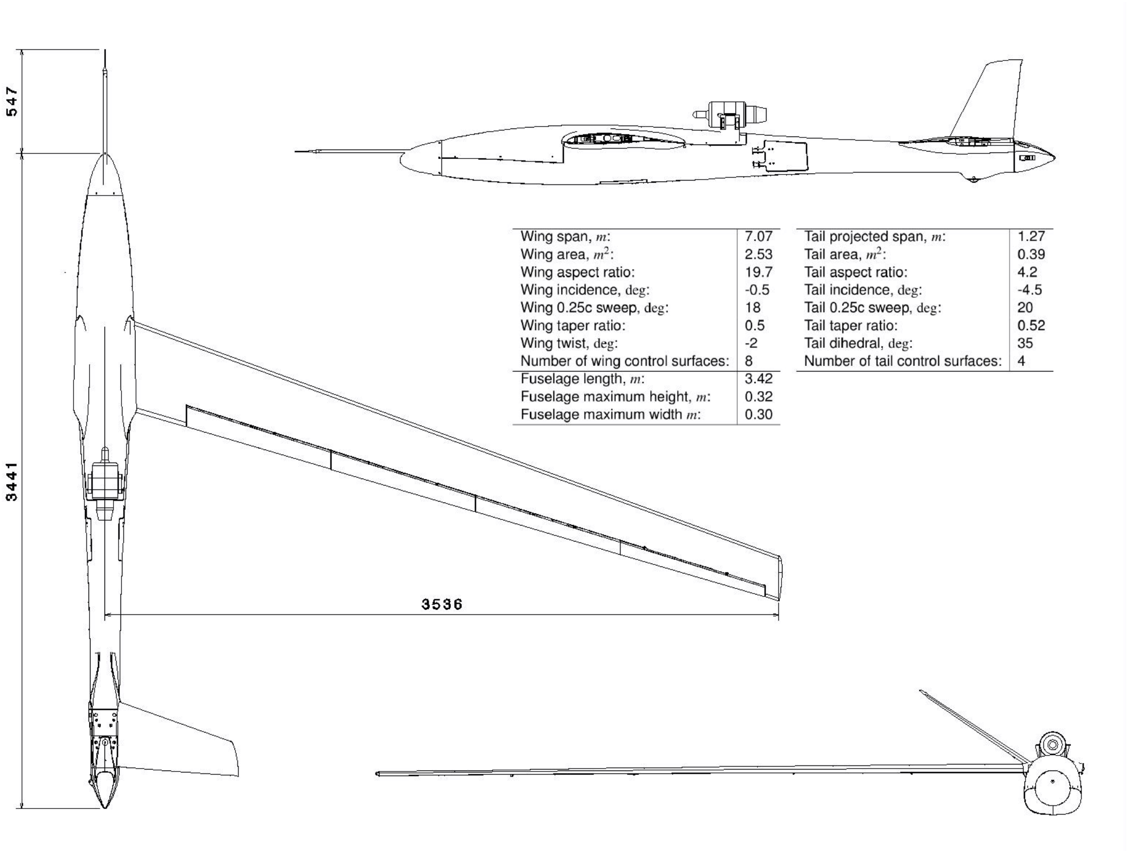
The aircraft has two control links. Control via two different transmitter brands was desired to decrease the risk of both transmitters failing together due to a common mode failure (either connectivity, electrical or mechanical). The main one is a Jeti DS-24 system which has an additional back-up receiver 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.

2.1.1 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.

2.1.2 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



Wing span, m:	7.07	Tail projected span, m:	1.27
Wing area, m^2:	2.53	Tail area, m^2:	0.39
Wing aspect ratio:	19.7	Tail aspect ratio:	4.2
Wing incidence, deg:	-0.5	Tail incidence, deg:	-4.5
Wing 0.25c sweep, deg:	18	Tail 0.25c sweep, deg:	20
Wing taper ratio:	0.5	Tail taper ratio:	0.52
Wing twist, deg:	-2	Tail dihedral, deg:	35
Number of wing control surfaces:	8	Number of tail control surfaces:	4
Fuselage length, m:	3.42		
Fuselage maximum height, m:	0.32		
Fuselage maximum width m:	0.30		

Figure 1: FLEXOP Subscale flight demonstrator. Note that the left wing and V-tail are excluded.

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 2.

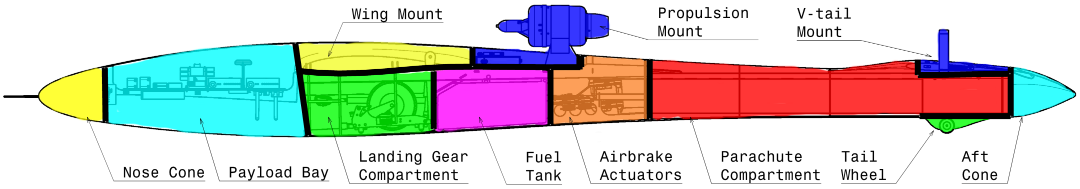


Figure 2: Interior layout inside the demonstrator.

2.2 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[1]. Figure 3 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

[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 3 you can observe the configuration modes and buttons for baseline mode functionality.

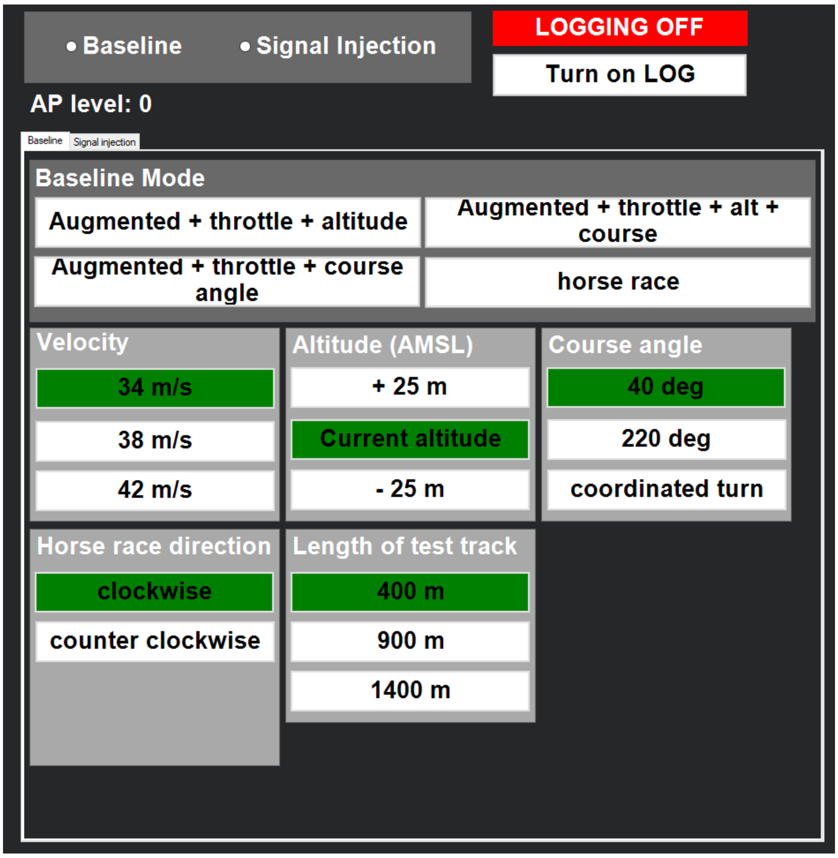


Figure 3: Mission Planner user interface

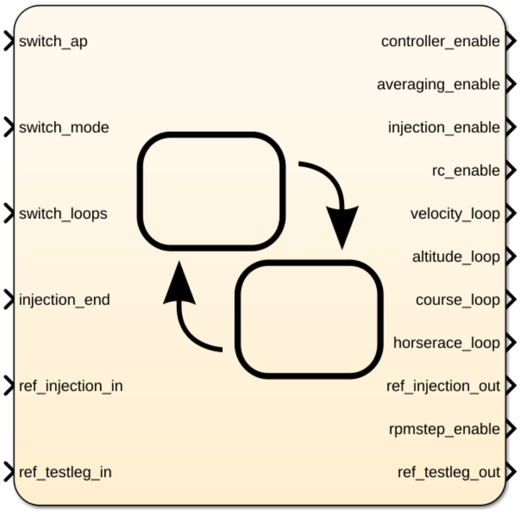


Figure 4: Matlab Simulink StateFlow Chart inside the autopilot

Simulink Coder. The mask of this state machine in Simulink can be seen on figure 4. 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 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.

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

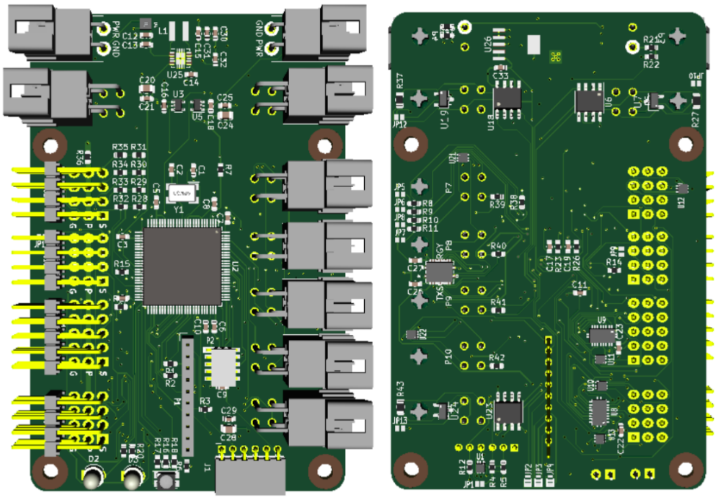
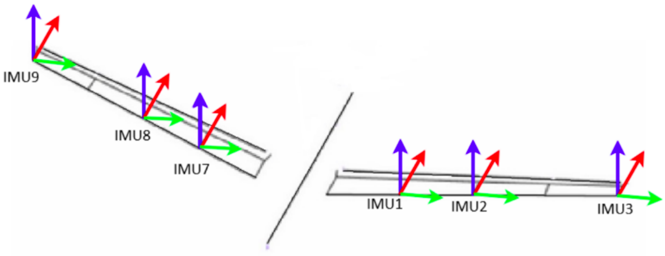
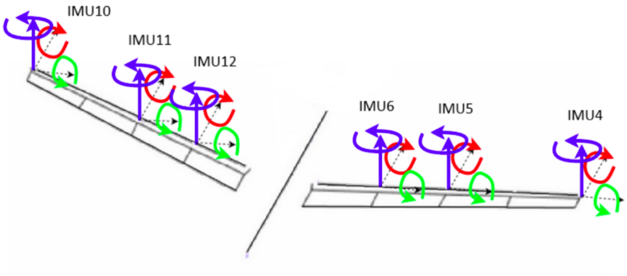


Figure 5: RX-MUX v2 render image

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 6: 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 7.

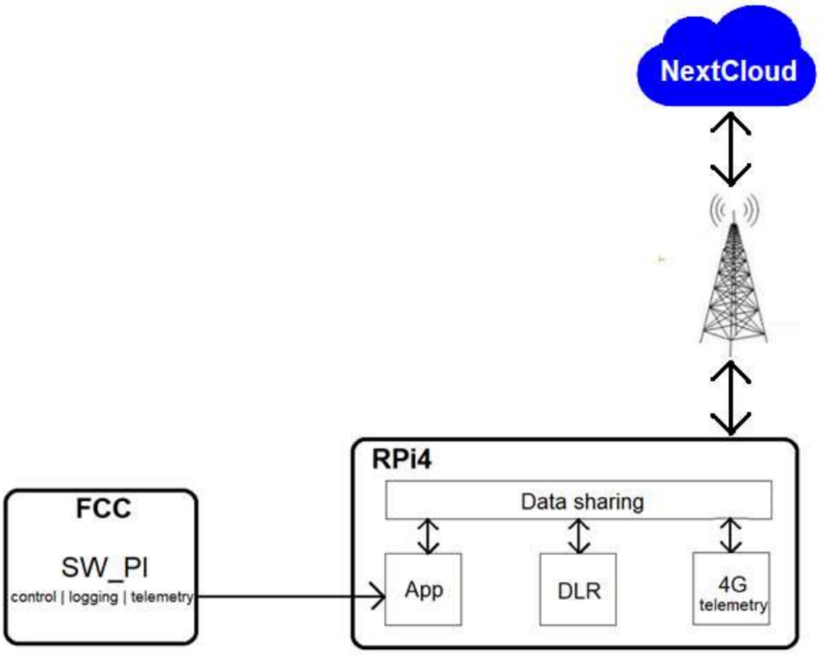


Figure 7: 4G telemetry concept

As it is visible in figure 7, the FCC sends data to the OBC-II via wired Ethernet connection. The ”App” in figure 7 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 8 which utilizes a Quectel 4G module to connect to the network.

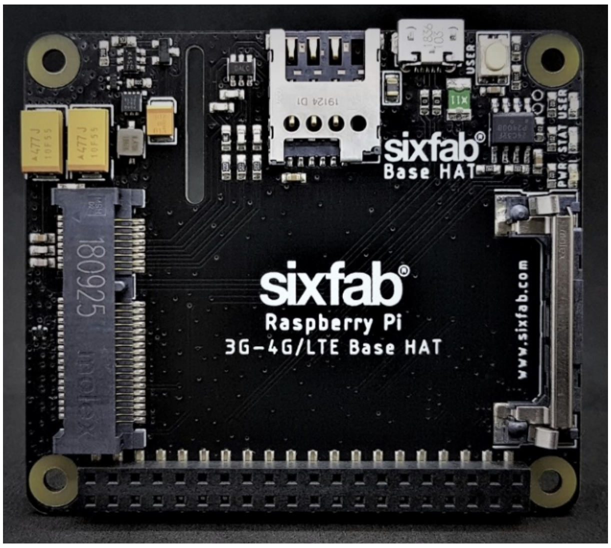


Figure 8: Sixfab 4G hat for the Raspberry Pi

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.

2.3 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.

2.4 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.

3 Ground Test Requirements

3.1 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 3.1.1.

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.

3.1.1 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.

3.1.2 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.

3.2 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 VV).

3.2.1 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

3.2.2 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.

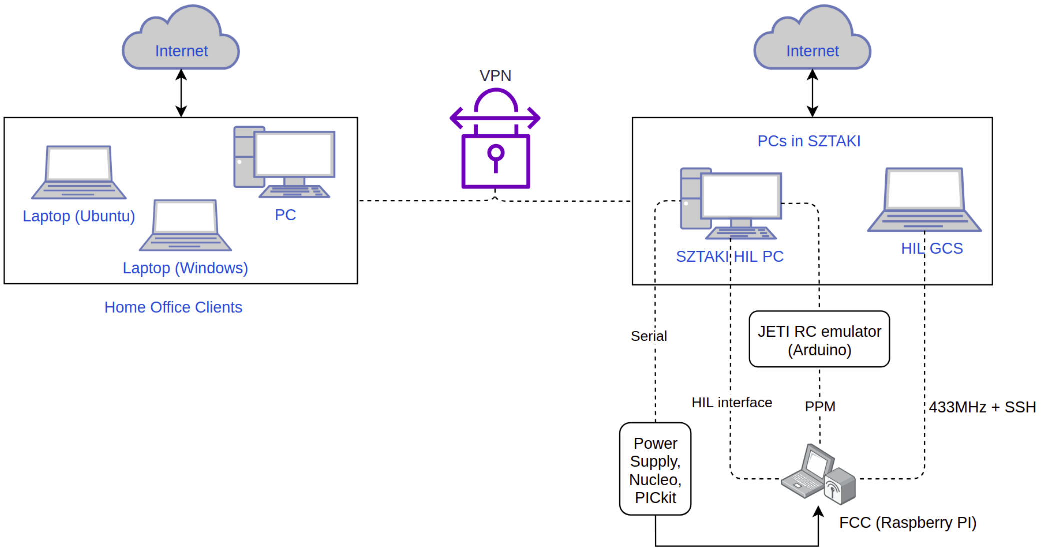


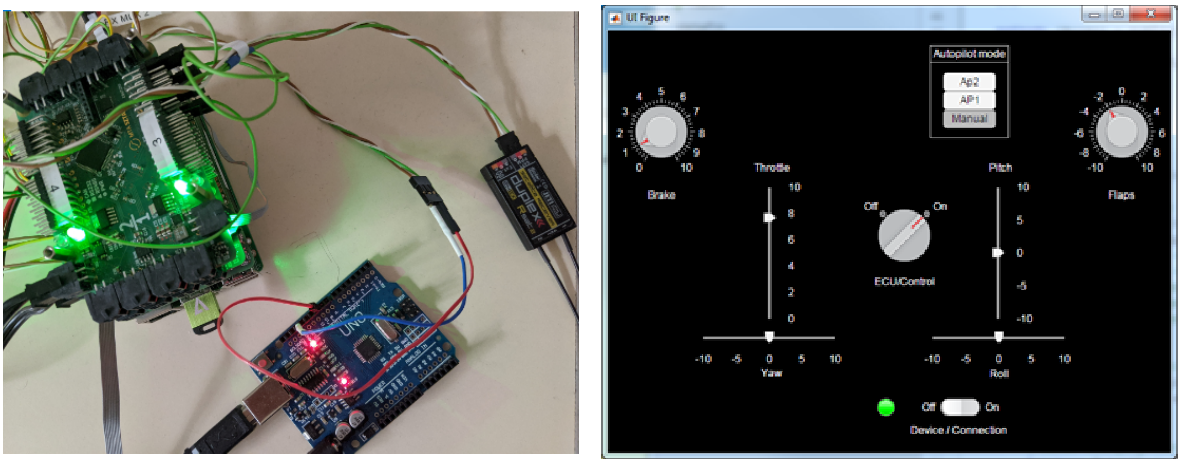
Figure 9: Remote configuration of SZTAKI HIL for home office access

- 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 10a.
- A client application in MATLAB was developed with JETI interfaces (Autopilot switch, joysticks), as seen on Figure 10b.
- Command-line interface is also provided in MATLAB

3.2.3 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 3. These buttons are controlling a state machine in the software (Figure 4). Basically the FLEXOP PaOT ran a SIL simulation on a target machine, connected to a visualization



(a) Arduino PPM generator device

(b) Matlab GUI for PPM generation

Figure 10: Remote HIL testing equipment

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 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.

3.2.4 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

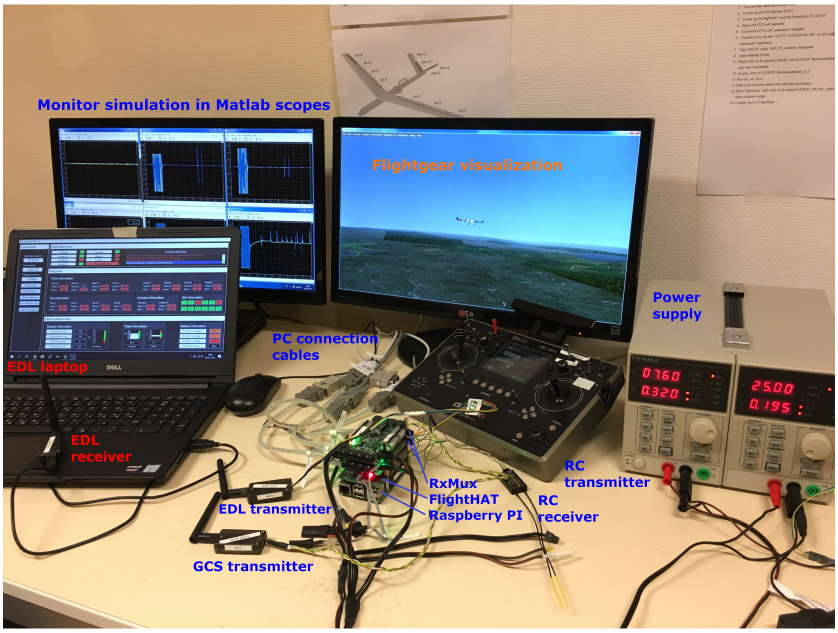


Figure 11: SZTAKI HIL which was sent to Munich

units as well. Snippets from the CAN and PWM input capture Simulink blocks can be seen in figures 12 and 13.

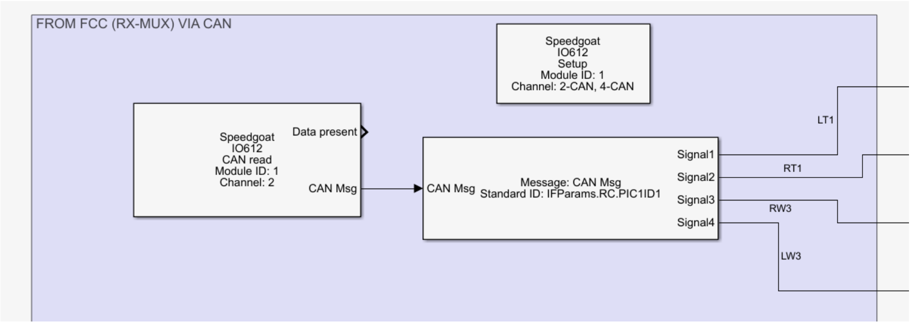

FROM FCC (RX-MUX) VIA CAN
Speedgoat IO612 Setup Module ID: 1 Channel: 2-CAN, 4-CAN
Speedgoat IO612 CAN read Module ID: 1 Channel: 2
Data present
CAN Msg
CAN Msg
Message: CAN Msg Standard ID: IFParams.RC.PIC1ID1
Signal1
Signal2
Signal3
Signal4
LT1
RT1
RW3
LW3

Figure 12: CAN input block of the Speedgoat Simulink model

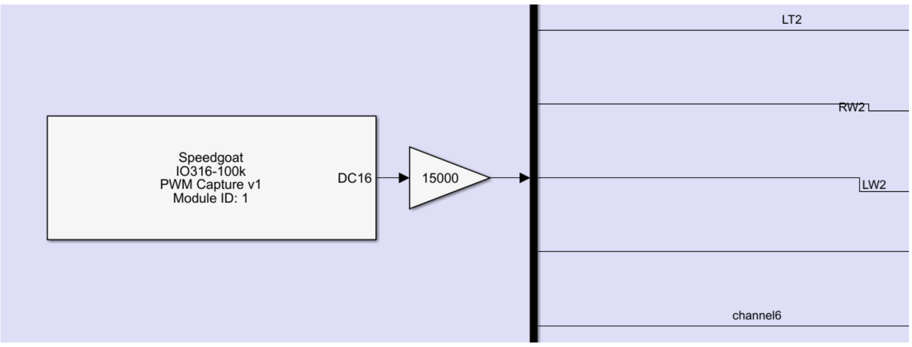


Figure 13: PWM input capture block of the Speedgoat Simulink model

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.

3.3 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.

3.4 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	$0 \leq T \leq 300$
Precision of Measurement	± 0.5
Duration of Measurement	$\geq 30\ minutes$
Sample Rate	$\geq 50\ samples\ per\ second$

4 Flight Test Requirements

4.1 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.

4.2 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 14 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 15 summarizes the proposed flight test plan schedule.

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

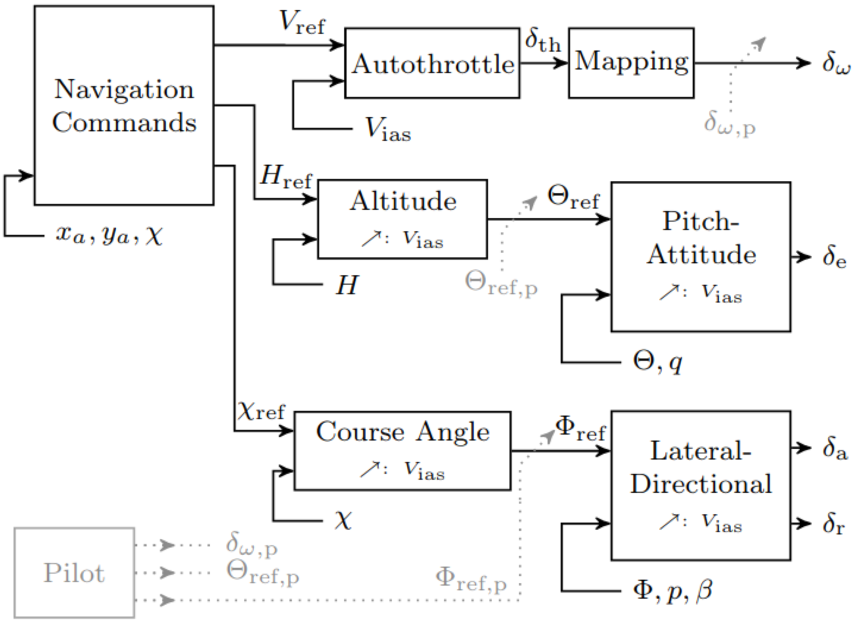
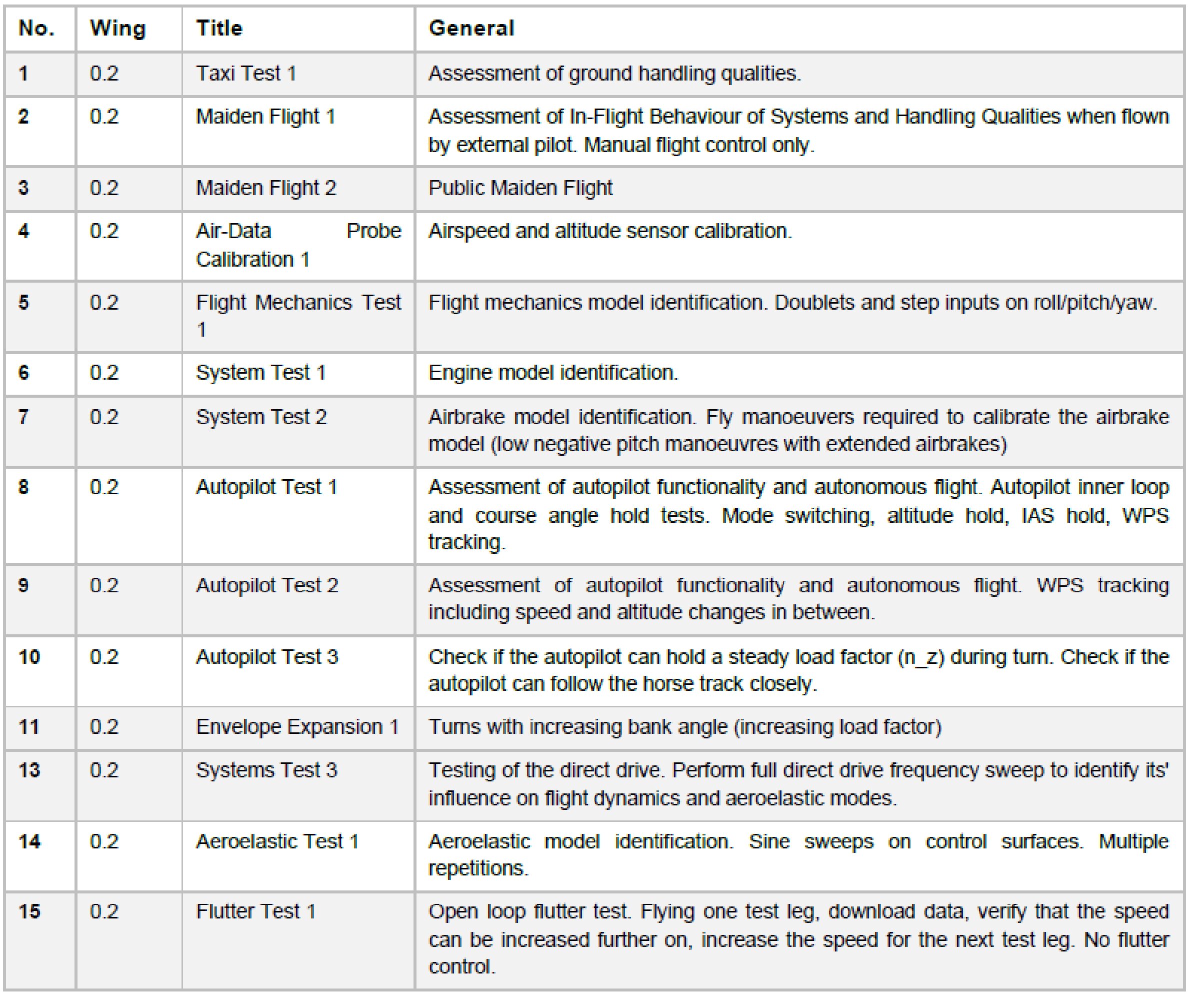


Figure 14: Architecture of the baseline controller

response of the engine. This non-linearity is included in the baseline control architecture (see Figure 14) 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 16). 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 provide valuable insights on the modeling and design methodology, formulated as formal metrics and incorporated in the integrated design.



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 15: Flight test plan related to the baseline controller [1]

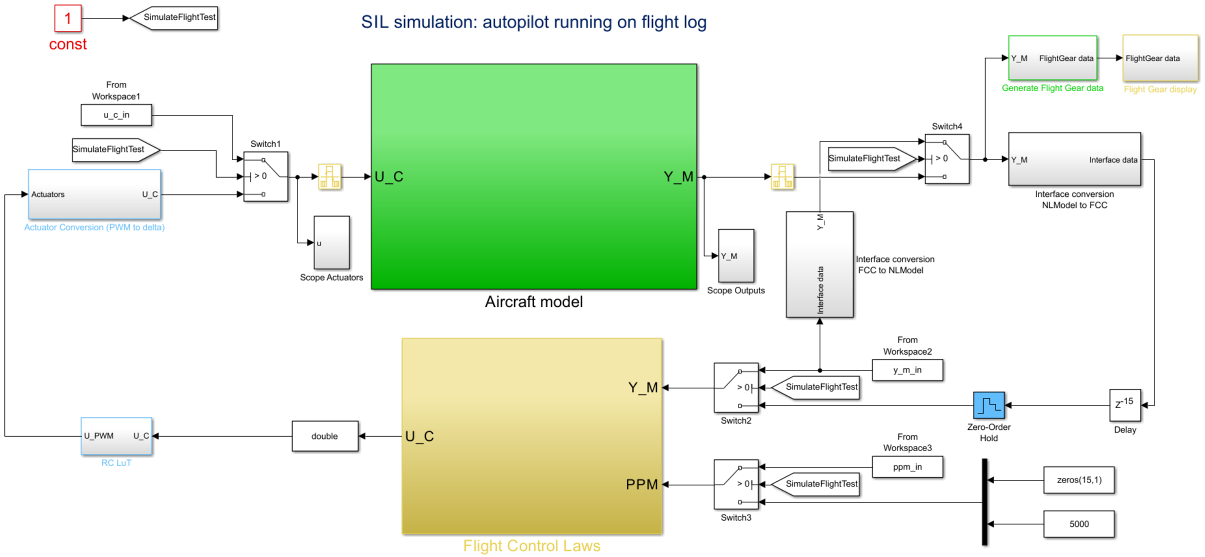


Figure 16: SIL simulation of the baseline controller based on flight log measurements and inputs

4.3 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.

4.4 Flutter control

The nominal flutter controller is developed by SZTAKI. The flutter controller [2] 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.

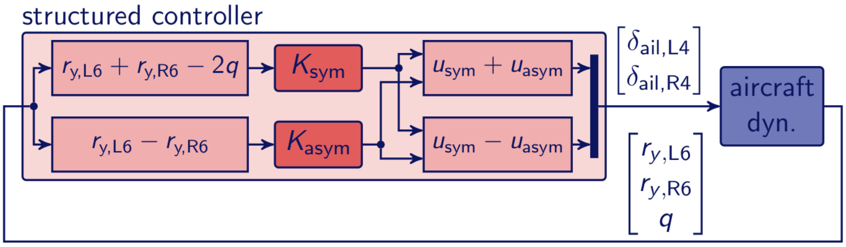

structured controller
$r_{y,L6} + r_{y,R6} - 2q$
K_{sym}
$u_{sym} + u_{asym}$
$r_{y,L6} - r_{y,R6}$
K_{asym}
$u_{sym} - u_{asym}$
$\begin{bmatrix}\delta_{ail,L4}\\ \delta_{ail,R4}\end{bmatrix}$
aircraft dyn.
$\begin{bmatrix}r_{y,L6}\\ r_{y,R6}\\ q\end{bmatrix}$

Figure 17: The structure of the closed loop with the flutter controller

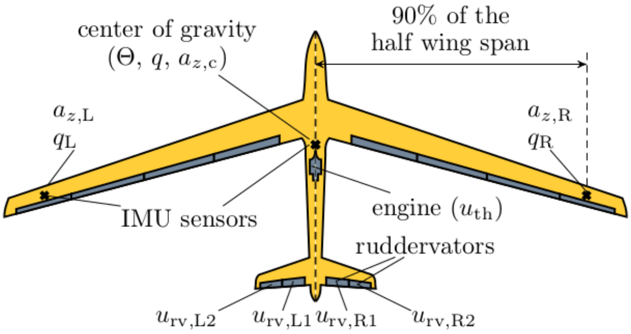


Figure 18: Sensors and actuators of the aircraft

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.

4.5 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 3.4).
- 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.

4.6 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 19 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).

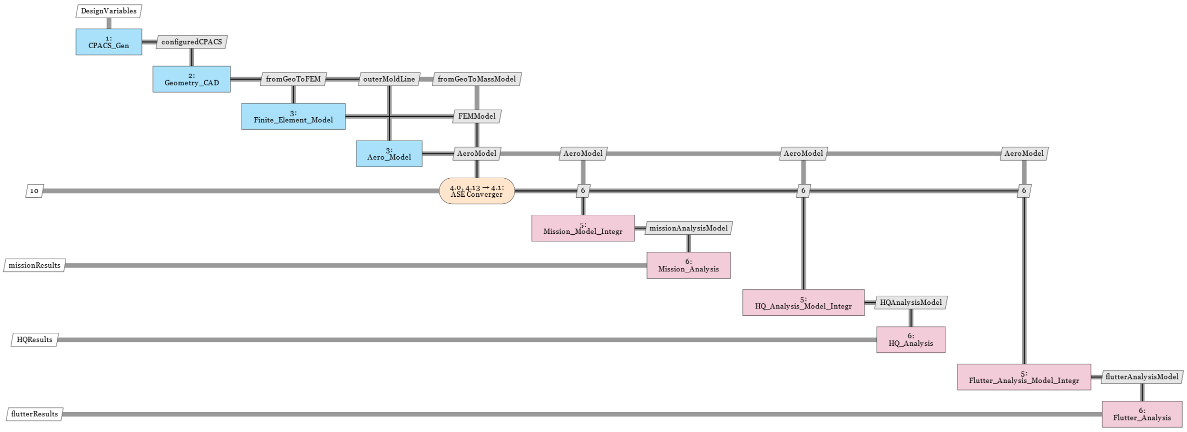


Figure 19: Toolchain steps illustration.

5 Conclusion and Test Plan

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 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.

6 Bibliography

[1] Charles Poussot-Vassal, Julius Bartasevicius, and Balint Vanek. Flipased D3.1 flight test programme – flight test phase #1. Technical report, SZTAKI, 2021.

[2] 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*.

[3] Balint Vanek. Flexop D4.10 release of the flight test results and models of the a/c for the community. Technical report, SZTAKI, 2019.