FliPASED
WWW.FLIPASED.EU

D3.3 Flight Test Programme – Flight Test Phase #2, Matthias Wüstenhagen (DLR), Keith Soal (DLR), Béla Takarics (SZTAKI), Szabolcs Tóth (SZTAKI), Mihály Nagy (SZTAKI), László Gyulai (SZTAKI), Tamás Luspay (SZTAKI), Julius Bartasevicius (TUM), Sebastian Koeberle (TUM)

"This document is part of a project that has received funding from the European Union's Horizon 2020

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: 05/03/2022	**Status:** On-Going
Due date: 15/11/2021	

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

research and innovation programme under grant agreement No 815058."

Table of contents

Glossary

AGL	Above Ground Level
ATZ	Aerodrome Traffic Zone
BLDC	Brushless Direct Current
CAN	Controller Area Network
CONOPS	Concept of Operations
EDBC	Magdeburg-Cochstedt Airport
EDL	Electronic Dispatch Logging
EDMO	Airport Oberpfaffenhofen
FCC	Flight Control Computer
FTO	Flight Test Operator
GCS	Ground Control Station
GUI	Graphical User Interface
GVT	Ground Vibration Test
HIL	Hardware in the Loop
HW	Hardware
IMU	Inertial Measurement Unit
LBA	National Aviation Authority of Germany
LRZ	Leibniz-Rechenzentrum
LTE	Long Term Evolution
MCT	Mission Control Technologies
MIMO	Multi-Input Multi-Output
PCI	Peripheral Component Interconnect
PID	Proportional–Integral–Derivative
PPM	Pulse-Position Modulation
RC	Remote Controller
SIL	Software in the Loop
SISO	Single-Input Single-Output
SW	Software
TMS	Thrust Measurement Sensor
UAV	Unmanned Aerial Vehicle
VLOS	Visual Line Of Sight
VPN	Virtual Private Network
VV	Verficiation/Validation

List of Figures

1 Executive Summary

This deliverable depicts the planned flight test programme 2 for flying the flutter critical -1 wing set with the demonstrator aircraft. A detailed flight test programme is established defining the test objectives, means of compliance, requirements on specific test procedures to be followed. The Flight test programme also specifies abnormal behaviour measures and quality gates.

The objective of this flight test campaign is to increase the flutter speed with control. Therfore, special care has to be taken by flying the demonstrator with the -1 wing as unstable symmetric and asymmetric flutter modes are expected within the flight regime in open-loop. Flying with this configuration is just possible below the flutter speed in open-loop or with a flutter controller switched on. For security reasons and also to judge the performance of the developed active flutter suppression system an operational modal analysis is performed in real time. It provides information on the actual frequency, damping and mode shapes of the aircraft modes.

So far simulations have proven the high potential of flutter control. Further steps would include a HIL testing of the proposed control laws and subsequently, testing in flight.

The deliverable is divided in three main sections. HIL testing and ground testing is explained in section 2. Section 3 depicts the general set-up and environment for flight testing the -1 wing. Finally, in section 4 the planned flight test programme is described.

2 Pre-Flight Tests

2.1 HIL Environment (SZTAKI)

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 interfaced 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). A detailed description is given in [3].

Although being used for years, the development of the old PC-based HIL system has not ended yet. The newest addition to the system is an environment which enables remote connection to the HIL PC and start the simulation from home office. This was crucial in the past circumstances for SZTAKI's employees as they were prohibited from going in to the institute.

The environment includes a remote control emulator which utilizes an Arduino board to generate PPM signal to the RX-MUX unit. So instead of using the actual RC receiver, the PPM signal generator provides the input. The Arduino board is connected to the HIL PC via serial interface and it can receive the PPM channel and PPM value as input. So for example, the autopilot switch can be toggled by sending either 7500, 8000 or 10000 to channel 16.

On the other hand, the HIL PC is only accessible in the internal network of SZTAKI, therefore a VPN network connection has to be established by the remote user. After that a remote desktop application named TightVNC has to be launched to access the graphical user interface of the HIL PC.

The current state of the HIL Simulink model can be seen in Figure 1.

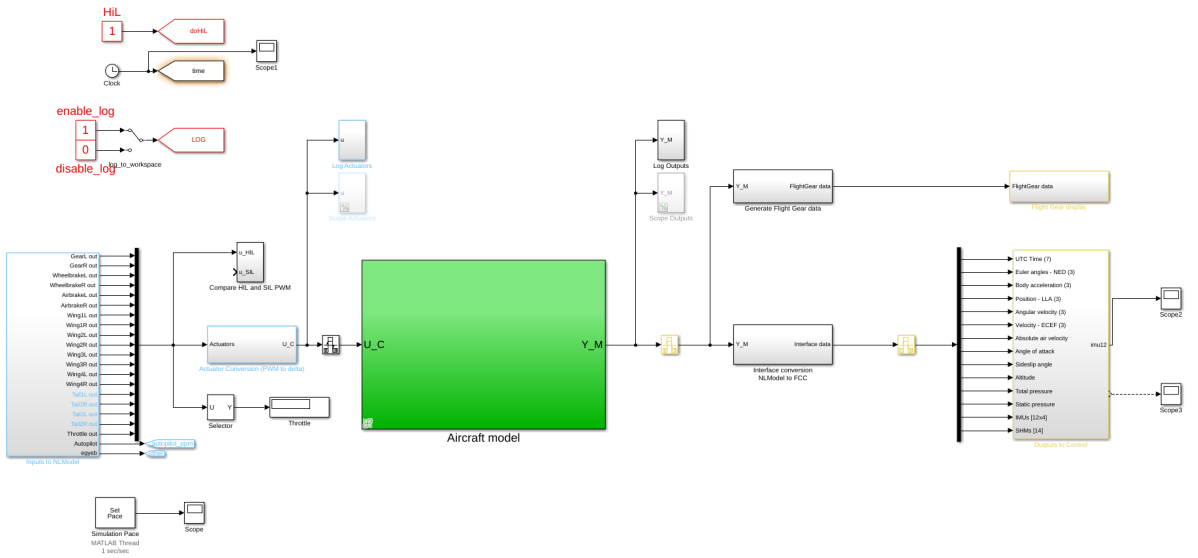


Figure 1: Current state of the HIL Simulink model

2.2 Ground Testing (SZTAKI)

The development of the FCC relies mainly on HIL tests. The software integrity and bugs can be fixed with the aid of this test environment at most of the time. Integration of new features always include HIL tests to prove that the criteria of the expected behavior is met. Automatizing of the tests are partially possible, but the evaluation of the results is currently a time consuming process with the growth of requirements.

The augmented mode and altitude hold, course angle hold can be tested in HIL, also the signal injection functionalities were evaluated during HIL tests.

Some features on the aircraft cannot be tested in HIL, and checked on the ground during system integration. The main field of these test is the cases which cannot be modeled, or when the test environment does not have any interface or capability to take them into account.

The following features need a special attention during a ground test:

- High-level autopilot functionalities
- Jet engine and turbine response
- Direct Drive and CAN servos
- 4G telemetry and particular OBC-II functions

For autopilot testing, several ground tests were made for benchmarking autothrottle or horserace functions. These involved some modified software components, because some sensor data was need to be fed to the FCC despite that the aircraft was on the ground. However the HIL environment does the exact same thing, the behavior of the turbine cannot be tested with HIL. For ground tests we prepare a test version of autopilot with only one difference compared to flight version: we give constant for the controller instead of measured speed. Horserace was tested in a similar way. The aircraft was rolled on its stand with wheels, and the control surface actuators were observed.

Direct Drive was tested with mounted on wings and using masses on the aileron to model aerodynamic load. Bandwidth tests were perform with sine sweep signals and step signals to ensure the safe operation.

The OBC-II, which is a companion computer for the FCC is a hub for several onboard measurement algorithms and devices. Currently, the integration of a 4G internet capable extension board is ongoing. The internet bandwidth and delays are also tested in the lab, but testing them on the airfield is also necessary.

3 Flight Test Set-up

3.1 Flight Test Environment (TUM)

As Mentioned in the D3.2 - Flight Test Report Phase 1, the rules for flying UAVs in Germany have changed significantly. Therefore, in preparation for the upcoming flight test season in 2022, a new extensive application for a flight permit had to be made.

The flight permit application required a new edition of Concept of Operations (CONOPS) that would be specifically written for this airport. This CONOPS document is available on request.

The first version of the application has been submitted to the National Aviation Authority of Germany (LBA) on 31st of August, 2021. The second version had to include some major changes to the application and the second version was submitted on 30th of September. After this stage, the feedback regarding the application was only received on 17th of December and was implemented in the third version of the application. The application was submitted on 13th of January and is currently under review.

Note that the application is only applicable to the Magdeburg-Cochstedt Airport (EDBC), as it was decided to be the main airport for the flight season 2022. However, acquiring a flight permit for Special Airport Oberpfaffenhofen (EDMO) should be easier and the application is already being compiled.

3.1.1 Preparations for flying at Magdeburg-Cochstedt Airport (EDBC)

To alleviate the probability of having the runway blocked by traffic, it was decided to apply for a flight permit at Magdeburg-Cochstedt Airport (EDBC).

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 2. The airport layout is seen in Figure 3. EDBC airport has

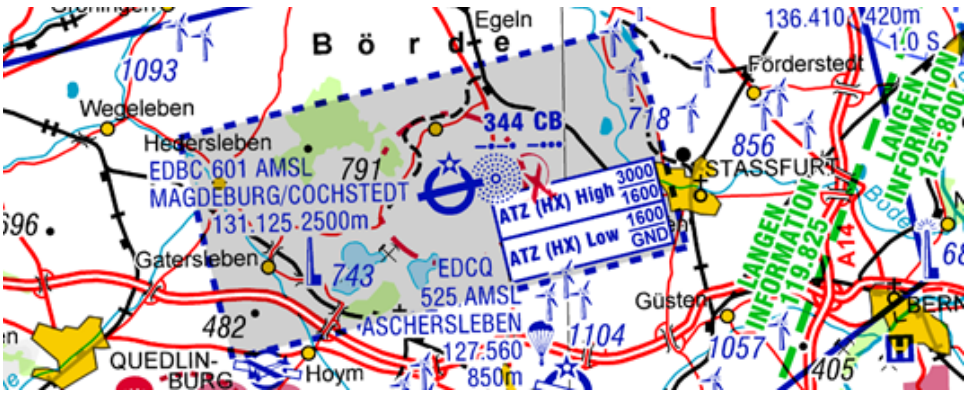


Figure 2: EDBC and the surroundings.

the possibility to close the airspace above it (ATZ (HX) Low). 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.

A ground risk buffer calculation was done for this altitude case. To estimate the ground risk buffer, a cus-

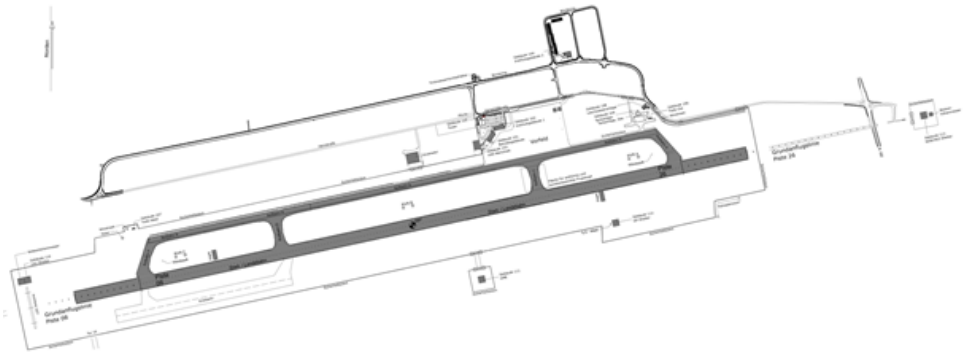


Figure 3: EDBC Airport layout.

tom software was used. The software was adjusted with the characteristics of the installed parachute. Simulations were done with the parameters mentioned in Table 1 and using the contingency volume buffer calculated above.

For the simulations here $H_{CV} = 305m$ was used. This results in $H_{FG} = 147m$.

The simulations resulted in ground risk buffer of $S_{GRB} = 194m$ for no wind conditions (Figure 4) and $S_{GRB} = 514m$ for maximum wind conditions (Figure 5).

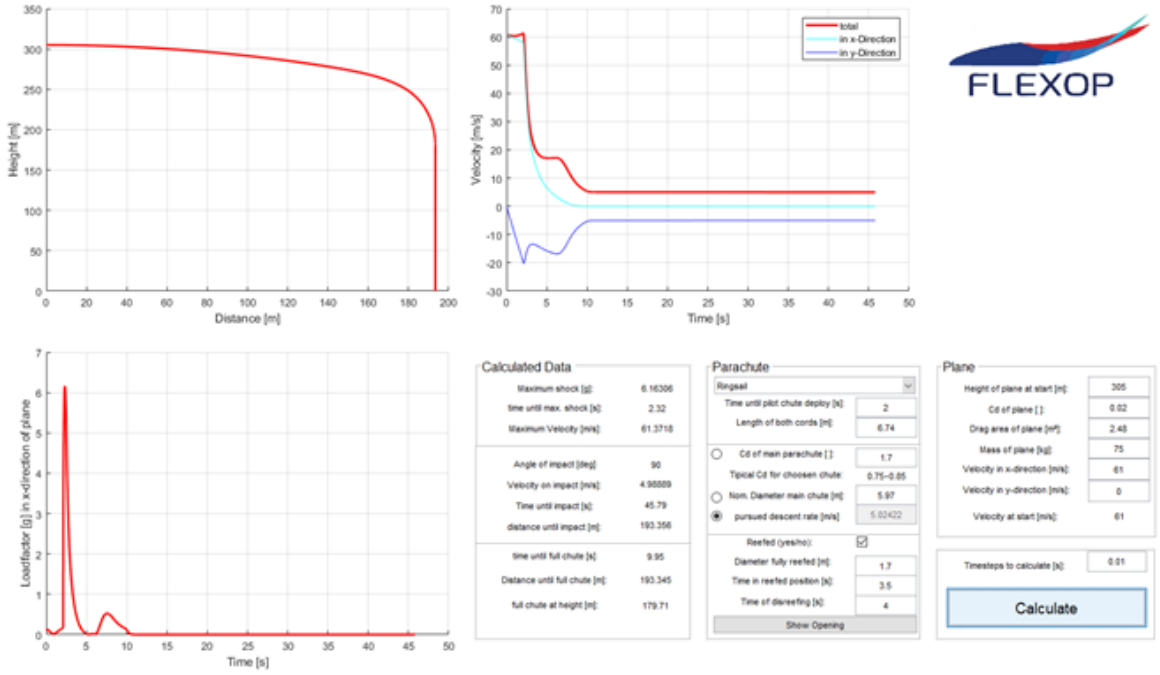


Figure 4: Parachute trajectory when released from 305m AGL, 0m/s wind case.

The flight geometry was built in the following manner:

- The optimal pilot position was identified (marked with a black dot).

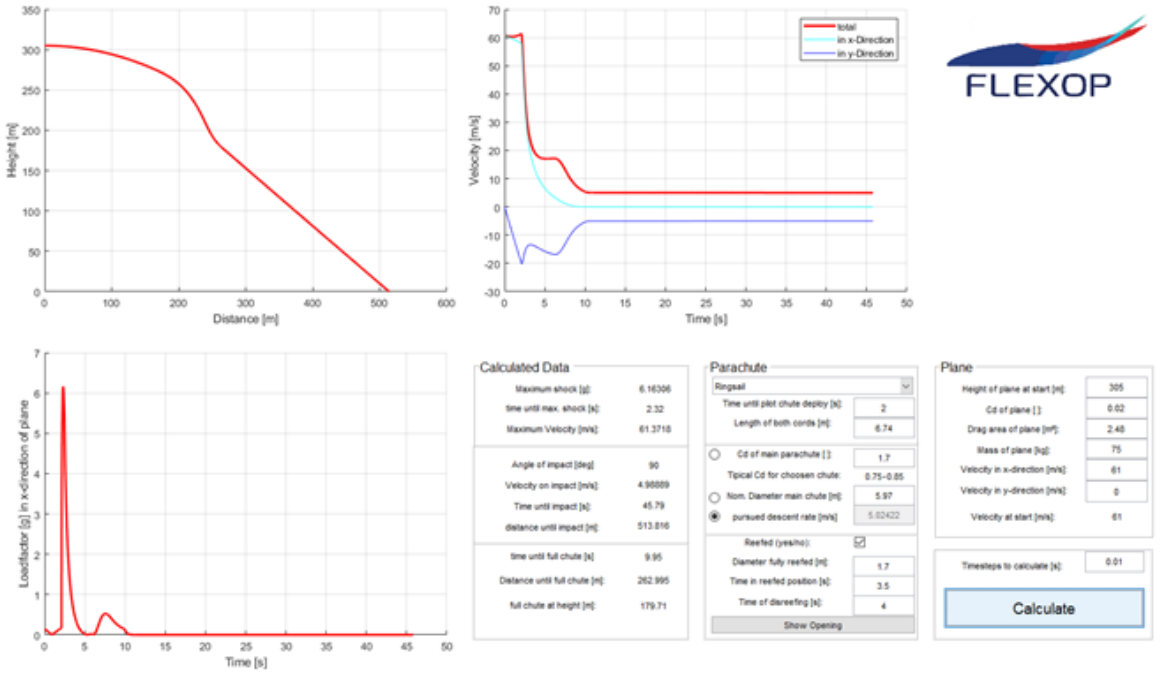


Figure 5: Parachute trajectory when released from 305m AGL, 7m/s wind case.

- A 1.5km (maximum VLOS distance) half circle was drawn around it. The area in the north of the pilot was selected to have some flexibility in case of a go-around procedure.
- Contingency and ground risk buffer zones were added.

The resulting flight geography required for the tests is shown in Figure 6. It covers:

- EDBC airport zone, which is closed to the public,
- Fields around the airport zone and
- A few local roads.

The distance to the UAV is maintained mainly by the pilot. However, FTO does see the distance via the telemetry and would inform the pilots well before reaching the 1.5km mark, making sure that a trajectory correction manoeuvre would not cross the border of the flight geography.

The T-FLEX flight box all lies within the ATZ (HX) zone. All the adjacent area is under the same ATZ zone where no manned aircraft are allowed.

3.1.2 Preparations for flying at Special Airport Oberpfaffenhofen (EDMO)

Due to the new rules for flight permits, an extension to the CONOPS mentioned above will have to be made in order to be allowed to fly at EDMO. The compiling of it is still in progress and continuous contact is kept with the National Aviation Authority of Germany (LBA).

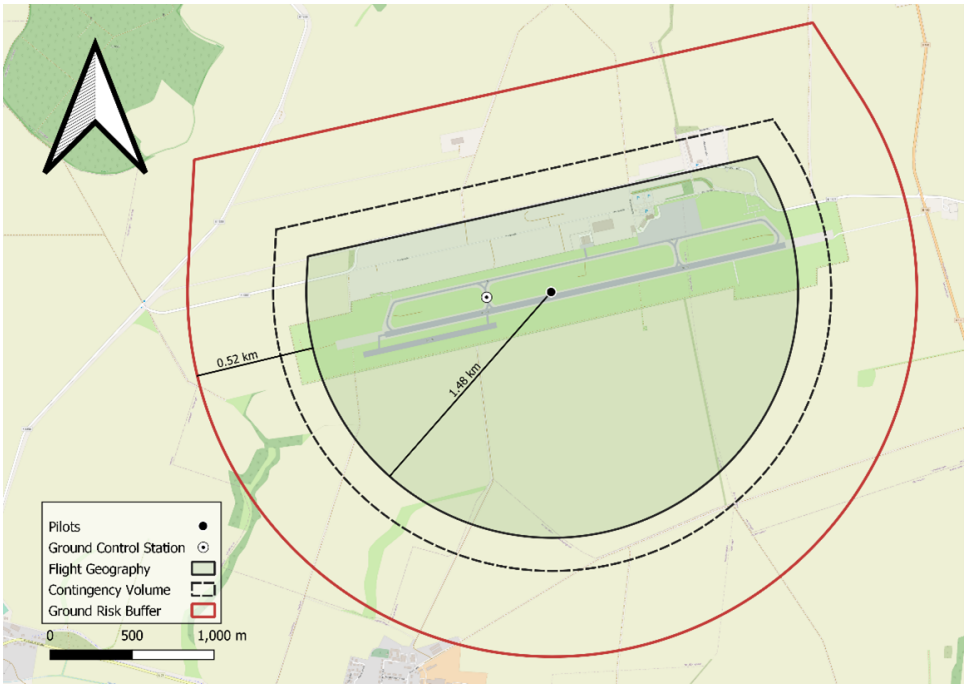


Figure 6: EDBC Flight Geography.

3.2 Demonstrator Set-up (TUM,SZTAKI,DLR-SR)

3.2.1 IMU configuration (TUM)

Based on results from the Ground Vibration Test (GVT) it was decided to re-configure the IMUs as well as instrument the empennage and fuselage with additional IMUs, as shown in Figure 7. This

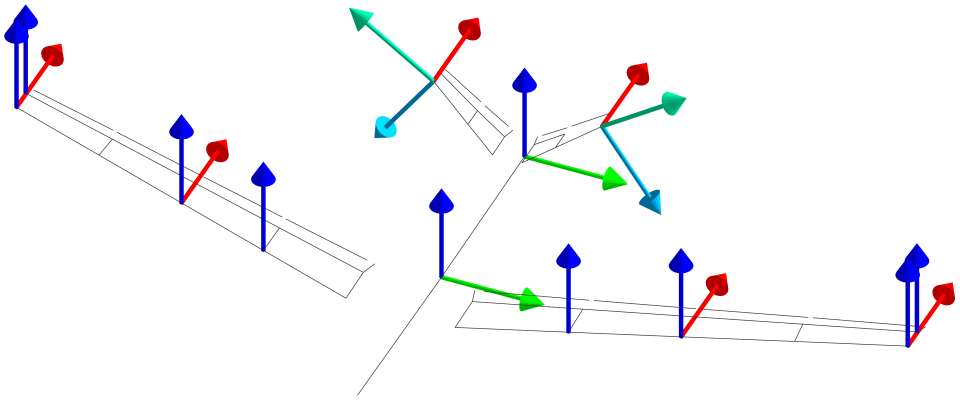


Figure 7: IMU re-configuration and additional instrumentation.

configuration increases the number of observable modes from 6 to 14 in the bandwidth to 30 Hz.

One IMU in each tail measuring acceleration in three directions will enable the identification of 4 V-tail modes in the bandwidth 17 Hz - 33 Hz. The IMUs have been located as far from the root as possible. The sensors are oriented in the local coordinate system to assist for practical installation purposes, and can be decomposed into the global co-ordinate system using Euler angles. For the purpose of modal analysis no angular rates are required on the IMUs in the empennage or fuselage, thus saving bandwidth on the CAN bus.

The IMUs in the wings were re-configured to include in plane measurements. Since they are glued into the wings and not accessible from outside, we took advantage of the bootloader on the flutterIMU boards to reprogram them via CAN bus. The new software is backward compatible with previous FCC software, therefore previous measurements are also reproducible. Not only the new wing will use this new mode, but the existing wings (-0, -1, -2) were also reprogrammed successfully. This allows the identification and tracking of the highly non-linear in-plane wing bending mode shown in Figure 8 and Figure 9.

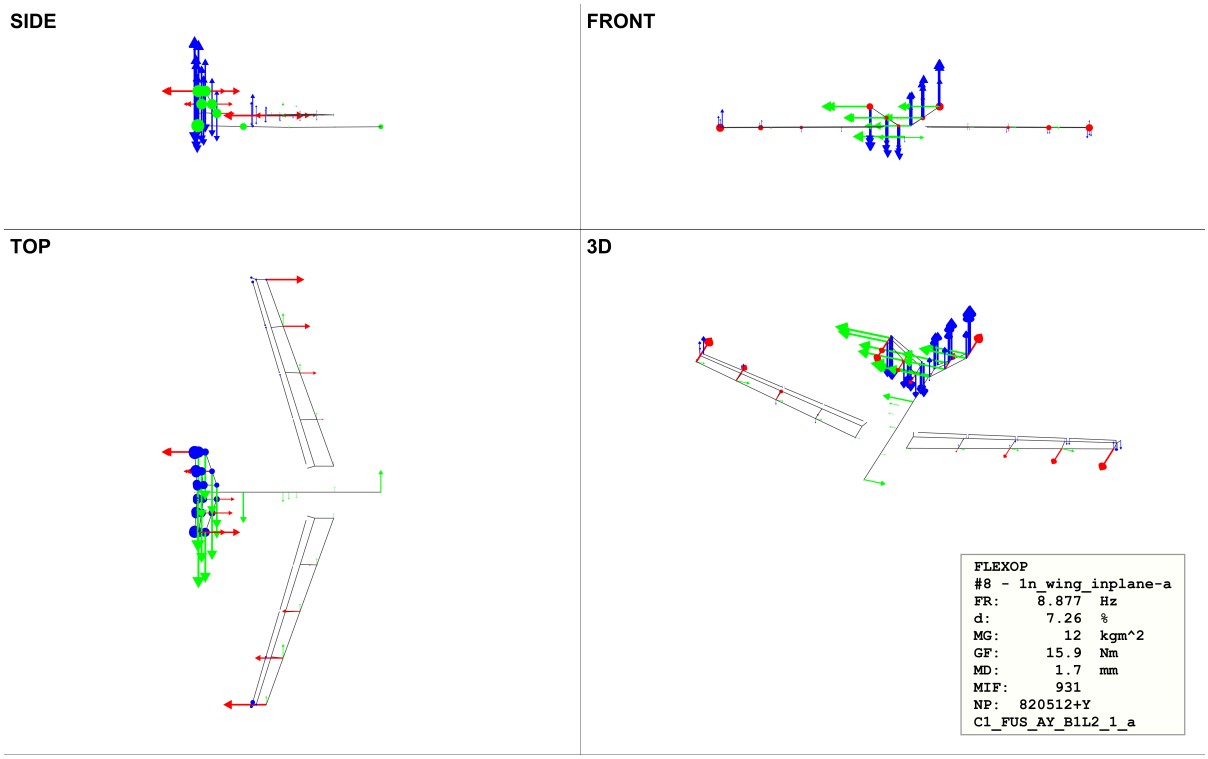


Figure 8: Four view plot of -1 wing in-plane bending.

3.2.2 -1 wing (TUM)

-1 wing was developed in FLEXOP project to demonstrate the flutter suppression techology. The design of -1 wing was restricted by several constraints. The flutter speed should be smaller than 55 m/s because of the limited flight path in airfield. All the test should be conducted in visual line of sight concept. Another significant constraint is the planform. In order to demonstrate the technology on the commercial aircraft later in the scale up task, the planform should be kept consistent with commercial aircraft. Therefore the planform of -1 wing was frozen to the parameters below:

The swept wings have a higher torsional loading and a higher torsional stiffness is needed for stability

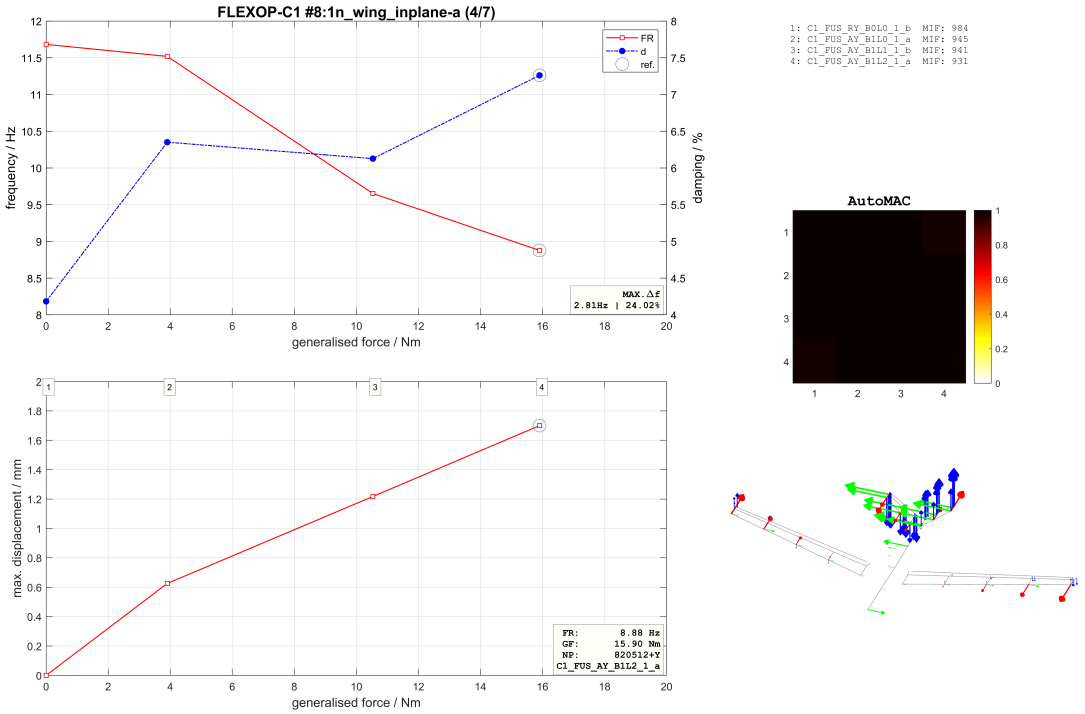


Figure 9: Non-linearity plot of -1 wing in-plane bending.

Table 1: Planform of -1 wing

Wing area, m^2:	2.5
Wing aspect ratio:	20
Wing taper ratio:	0.5
Wing sweep, deg:	20

reasons. Influence on flutter frequency is highly driven by the wings torsional stiffness. Lowering the wing torsional stiffness has a high impact but stands in a strong conflict with structural, aerodynamic and aeroelastic stability of the wing and therefore with the operational aspects [1].

In order to fulfill all the requirements, different structure concepts were investigated and parameter studies were conducted.

The -1 wing was final designed with a combination of lower torsional stiffness and extra flutter tuning masses.

3.2.3 Flutter mass (TUM)

To modify the dynamic aeroelastic behavior, the wing has 3 structural design degrees. These are additional masses in the mass bay, lever arm of the external mass and the weight of the external mass itself. Based on the necessary masses and the available design space, the mass bay is located between the main and the rear spar at 85 % of the semi-wing span. The main effect will be applied to the bending modes of the wing. The chosen design has the advantage, that for test reasons the additional masses can be taken out of the wing, what will stabilize the dynamic aeroelastic behavior. As a second part, a torsional bar and flutter mass can be mounted to the wing, what will mainly influence the torsional wing modes due to different leaver arm lengths and weights [1].

3.2.4 Flutter Stopper mechanism (TUM)

In April 2020 it was decided to implement additional safety features for the flutter tests. The discussion resulted in a Flutter Stopper development.

Flutter Stopper makes use of the fact that for -1 wing to flutter, weights need to be mounted aft on carbon rods next to the flutter actuator (Figure 10). Without these weights the aircraft flutter speeds are

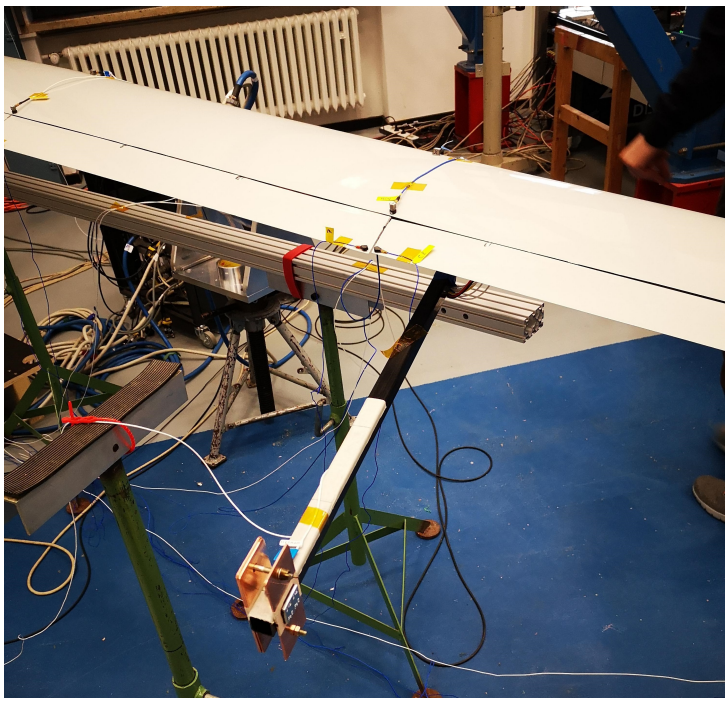
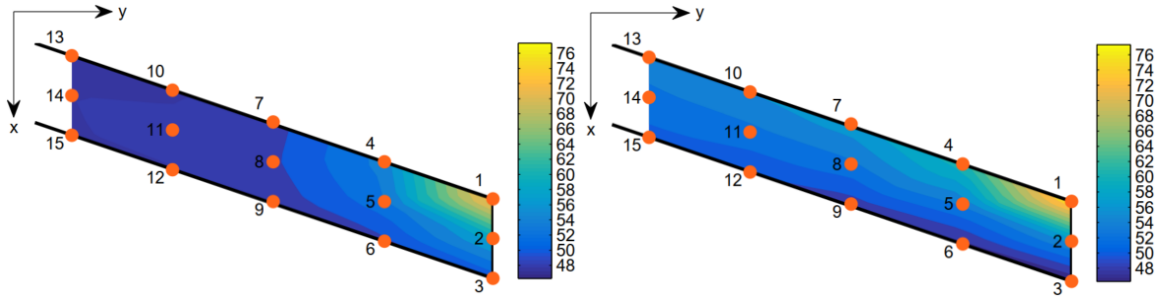


Figure 10: The weights required for flutter were simulated during the GVT campaign of the -1 wing at DLR.

significantly higher. Therefore it was decided to construct a mechanism which, when triggered, would move the weights forward increasing the flutter speeds. How the flutter speeds are affected with respect to different mass positions is exemplary shown in Figure 11 for various actuator positions. Based on this



(a) Symmetric flutter

(b) Asymmetric flutter

Figure 11: Flutter speeds [m/s] for different actuator positions

study the actuator positions were selected previously in the project. The same effect of an increasing flutter speed would also be expected with the described flutter stopper mechanism.

The mechanism consists of the flutter weight, a trigger mechanism, spring tensioner, buffer and a weight catcher (Figures 12 and 13). As the mechanism is triggered, the energy stored in the preloaded spring

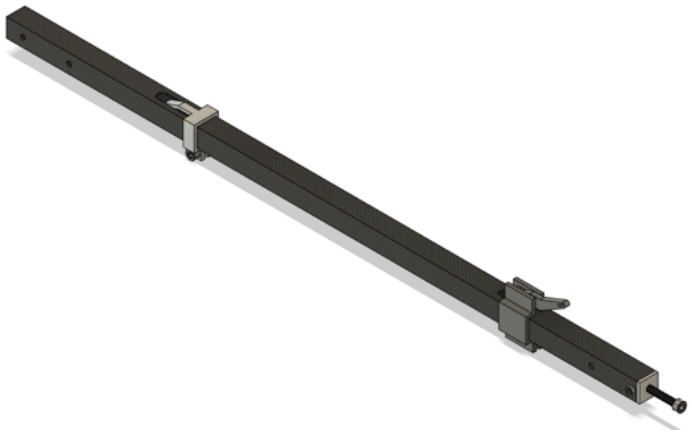


Figure 12: Rendering of the Flutter Stopper mechanism.

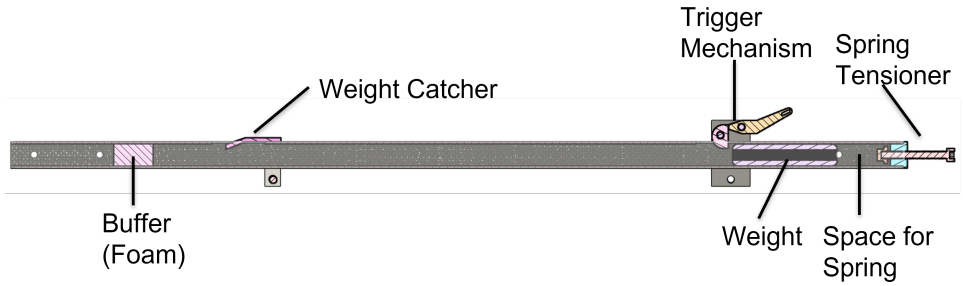


Figure 13: Construction of the Flutter Stopper mechanism.

is released and the weight is shot forwards. When it hits the buffer, weight catcher makes sure that the weight stays in place during the remaining of the flight.

As the mechanism was not the main priority for TUM, it is still in development. However, multiple iterations have already been tested with different spring, trigger and weight combinations. The final prototype is currently in production (Figure 14).

3.2.5 Direct Drive (SZTAKI)

During the Flight Test Phase 2, the flexible -1 wing will be tested with flutter controller described in the next section. The Direct Drive (on Figure 15), a high-bandwidth and high torque actuator acts on the outermost aileron control surface, as seen on Figure 16. This servo drive solution could demonstrate active flutter suppression to increase safe flight envelope.

Recently the RX-MUX unit of the FCC was redesigned to improve thrust-worthiness and capabilities of this component. The role of the RX-MUX is to handle all actuator outputs, including the Direct Drive. The former limitations of this unit were eliminated. For example, jitter occured in CAN communication. The redesign of RX-MUX unit brings a lot of improvement to the FCC, and the integration of the new features is continuous.

The inner state machine of the Elmo Gold Twitter could be handled properly, including switching on the

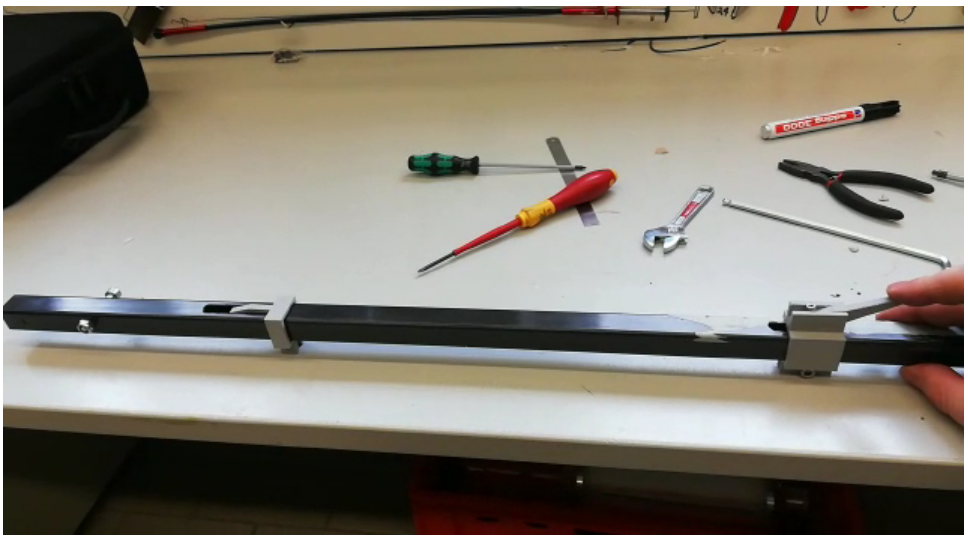


Figure 14: Testing of the Flutter Stopper mechanism. Different springs as well as different weights have been tested.

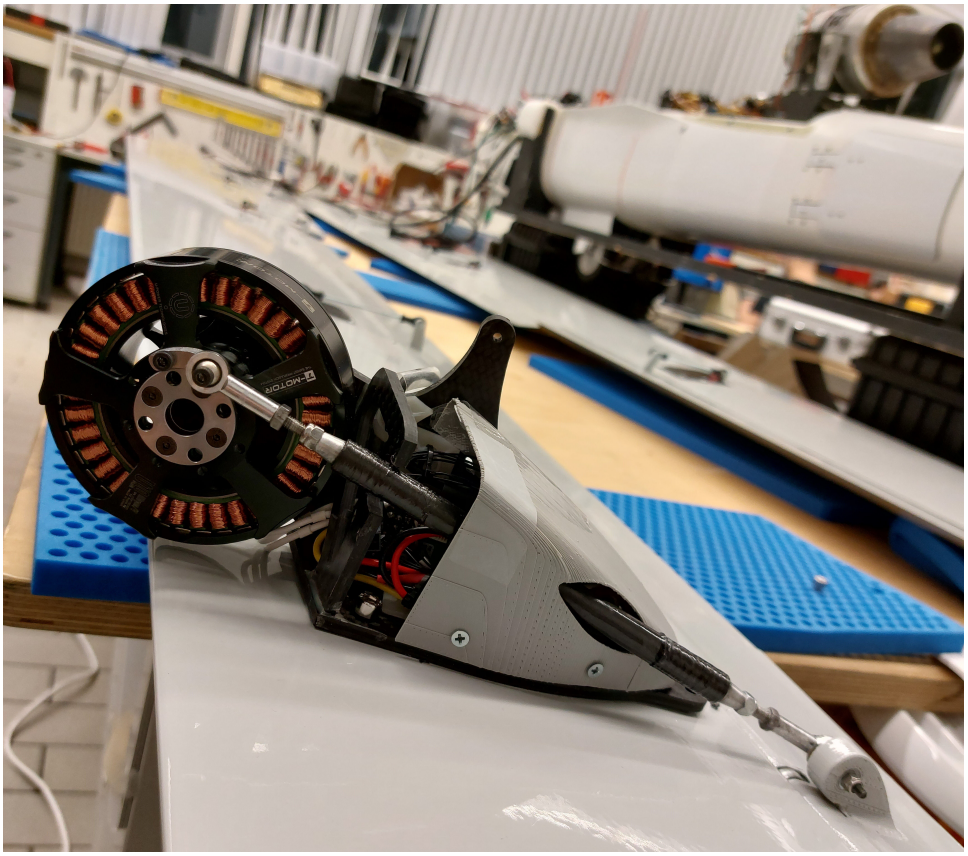


Figure 15: Direct Drive mounted on -0 wing for system integration

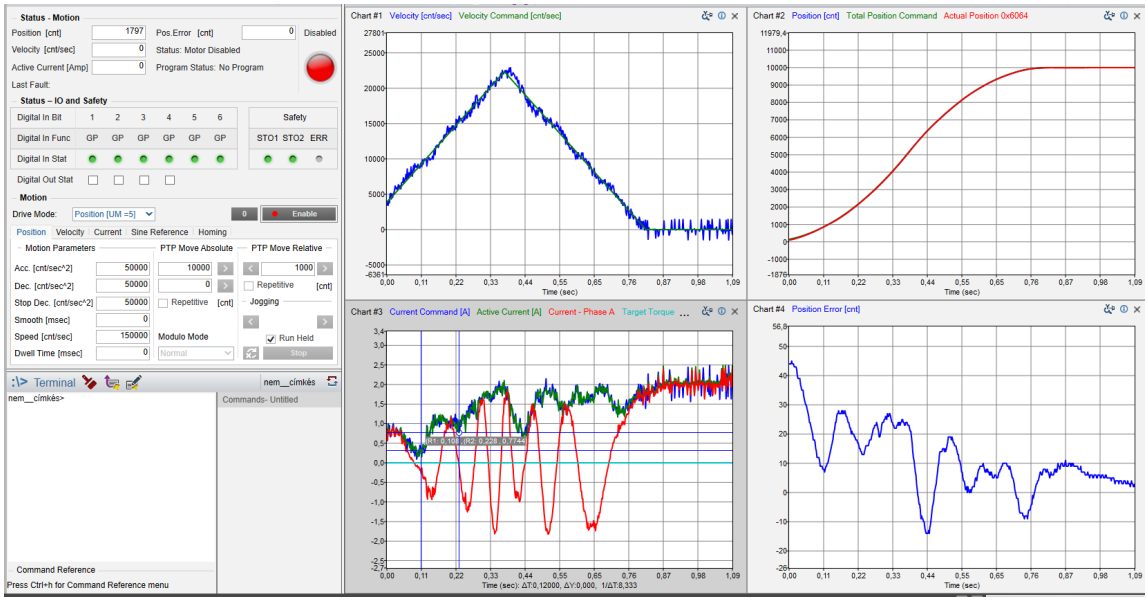


Figure 16: Direct Drive measurements under load

motor, set speed and acceleration limits, setpoints for moving the Direct Drive and to hold its desired position. Furthermore, diagnostic data queries are managed efficiently, with low CAN bus load. This is ensured by using a CANopen library for constructing the necessary messages, and some variables and status information is sent regularly by the Elmo Gold Twitter controller. The supply voltage, current, the position and velocity of the BLDC motor is queried and logged on the FCC. The implementation of these variables to the telemetry channels is in progress. Using the openMCT framework described in Section 3.4, the desired variables can be visualized easily.

The motion limits (e.g. acceleration) can be fine-tuned by the RX-MUX unit for a more aggressive or for a robust behavior. The Elmo Gold Twitter's cascade PID controller for position, velocity and current control is tuned with an inertia attached to model flap and aerodynamic effects.

Measurements with inertial load (using 2x100g Pb weights, one on the BLDC motor, and another at the end of a $0.5\,\mathrm{m}$ rod, with results on Figure 16) proved the stiffness of the actuator. Fine-tuning and the seen visualization of the motion is possible via the Elmo Application Studio II, with serial connection to the Elmo Gold Twitter motor controller. Some more measurements needed for determine the exact bandwidth attached on the wing, too.

At the last integration visit at TUM, SZTAKI installed the Direct Drive and made the wiring in the -1 wings (Figure 15). The actuator was attached to the flap, and some calibrations were made[1].

3.2.6 Flutter control - DLR (DLR-SR)

One of the flutter controllers was developed by DLR. It consists of two SISO controllers as symmetric and asymmetric flutter are found to become unstable within the flight envelope. Flutter is suppressed by deflection of the most outer ailerons shown in green in Figure 17. Based on a $\mathcal{H}_2$-optimal blending approach the pitch rate and vertical accelerations of all 13 IMUs installed in the fuselage and the wings are combined in order to determine the aileron deflection. A robust flutter controller clearly enlarges the flutter speed. As shown in Figure 18 the closed-loop flutter speed of symmetric flutter is increased from $52\,\mathrm{m/s}$ up to $65.5\,\mathrm{m/s}$, while the asymmetric flutter speed is increased from $54.5\,\mathrm{m/s}$ up to $70\,\mathrm{m/s}$.

[1] See the Direct Drive controlled by the JETI transmitter: `https://youtu.be/-Jd9J3w3-NE`

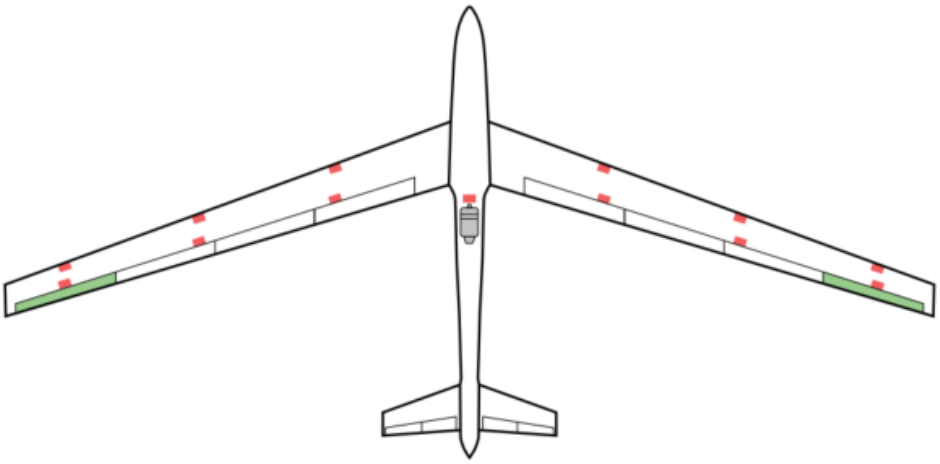


Figure 17: Aircraft with ailerons (green) and IMUs (red) used for flutter suppression

Further information on the flutter control synthesis can be found in [2].

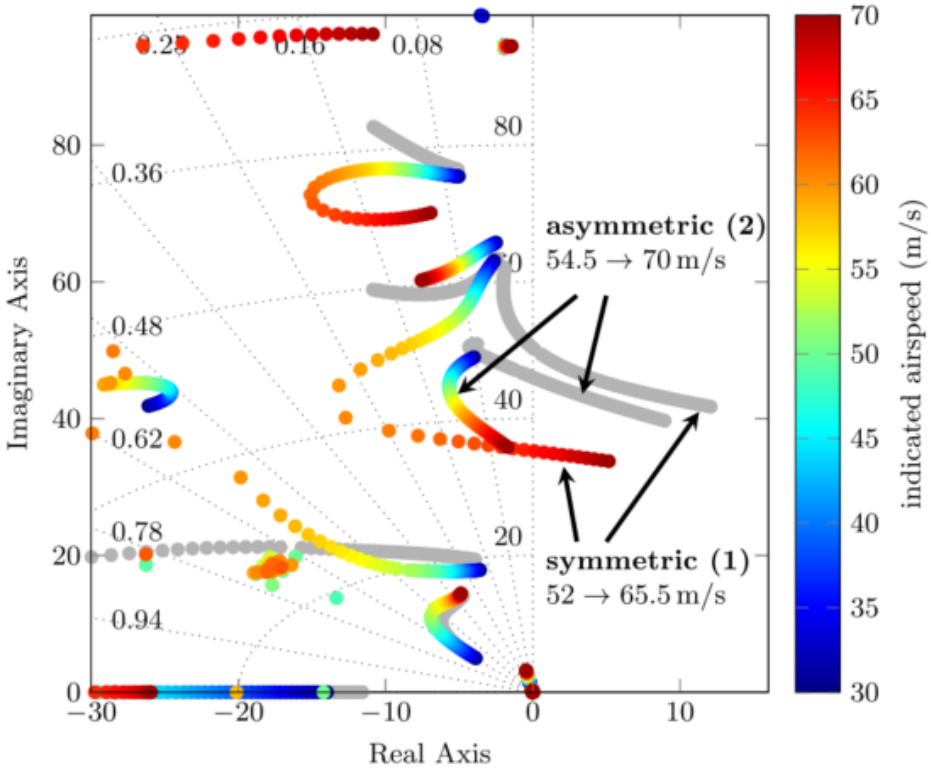


Figure 18: Open-loop flutter poles (grey) compared to closed-loop flutter poles at various flight speeds

3.2.7 Flutter control - SZTAKI (SZTAKI)

The flutter controller [4] 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 (Figure 19). The

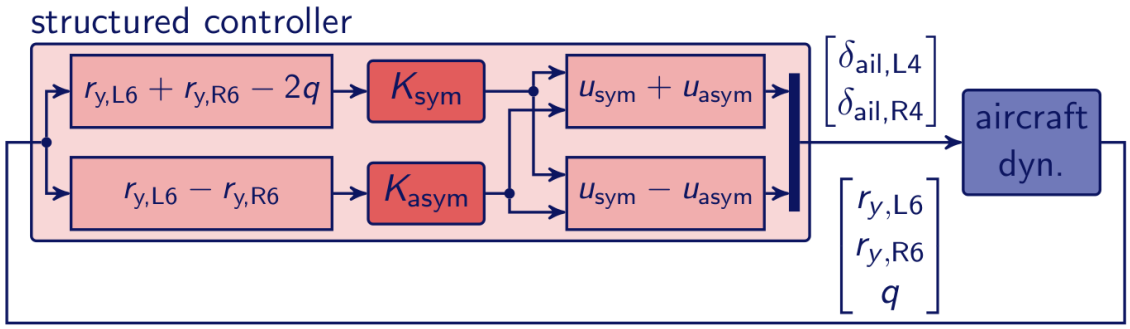


Figure 19: The structure of the closed loop with the flutter controller

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. The sensors and actuators used by the flutter controller are shown in Figure 20.

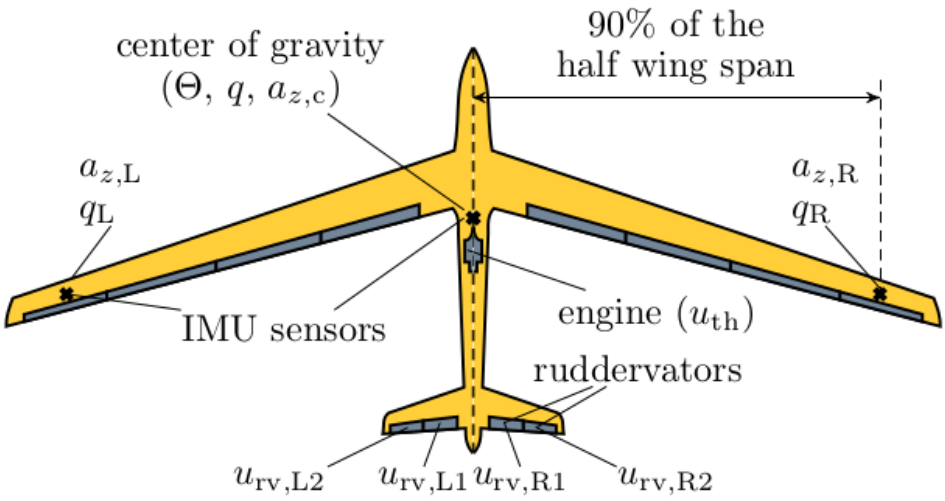


Figure 20: Sensors and actuators of the aircraft

3.2.8 Baseline controller (SZTAKI)

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 21 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.

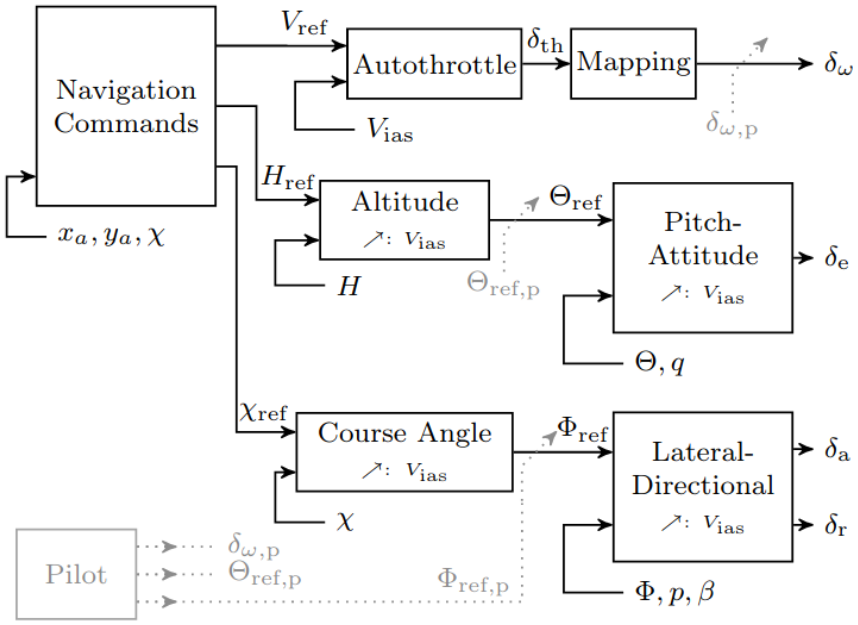


Figure 21: Architecture of the baseline controller

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.

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 22). 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.

3.3 Flight Test Crew (TUM)

The Flight Test Crew will be made up from members of TUM. Last year, TUM had 8 people available for the different roles of flight test crew (3 pilots and 5 engineers). Due to a change in personnel, 2 members have permanently left the team and one will not be available for a big part of the summer. Therefore 3 new members were recruited and are currently going through the training program.

The training program consists of the following phases:

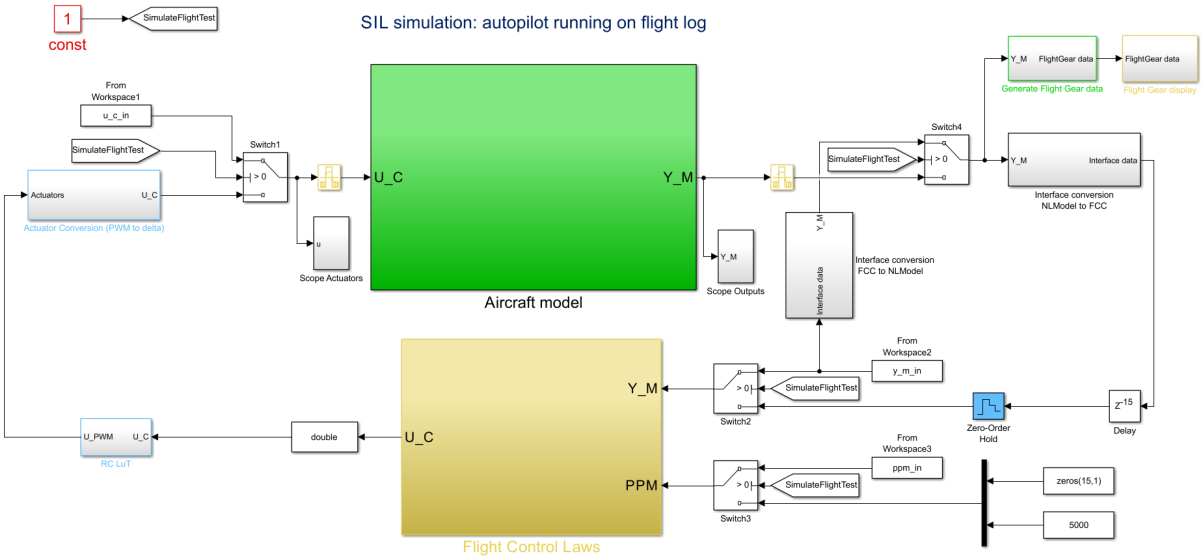


Figure 22: SIL simulation of the baseline controller based on flight log measurements and inputs

- 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.
- After that, training for specific roles will be done. Daniel Teubl will prepare a handover for the role of a Flight Test Engineer. Sebastian Koeberle will prepare a handover for the role of a Flight Test Operator. Julius Bartasevicius will prepare a handover for the Flight Test Manager role.
- 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.

After the training program is finished, there will be 3 pilots and 5 engineers available full time, plus one engineer available partially.

Pilots will do a separate training to refresh the skills flying similar-sized UAVs.

3.4 Ground Station (TUM,SZTAKI,DLR-AE)

Environment (TUM)
The Ground Control Station van is planned to be outfitted with a central server, acting as a central hub for telemetry data, the central Open MCT-server host, as well as provides internet access using an LTE-router as a gateway. Furthermore, the already existing antenna tracker was outfitted with functionality to automatically find its orientation. Expected advantages of the new setup can be summarized as:

1. Increased reliability and faster setup times.
2. Redundancy of telemetry data paths.
3. Redundancy of telemetry data displays.
4. Addition of data displays facilitating data evaluation during test flights.
5. Increased inter-connectivity and interaction among partners.

In the following, the individual points are elaborated:
1. Increase reliability and faster setup times: The telemetry server is setup with the operating system *Ubuntu 18.04 LTS*, which promises sufficient support, while the operating system's maturity allows for reliable operation. The versatility of the UNIX-based system and the dedication to the telemetry server role allows for adjustments like automatic telemetry module recognition and script-based setup automation that reduce the error rate while reducing the times needed to setup the Ground Control Station in the field.
2. Redundancy in telemetry data paths: The antenna tracker is planned to receive all the data contained in the 433 MHz datalinks (i.e. MAVlink, EDL, and modal flutter analysis data) making it accessibile within the network. Therefore a loss of a single data link or even shutdown/crash of a single computer will not result in loss of all telemetry data. This also enables the hosting of additional OpenMCT telemetry server instances.
3. Redundancy of data displays: Since program crashes of the primary MissionPlanner software has in the past lead to problems during flight tests and in one occasion lead the execution of an expedited landing, the new setup is expected to provide greater robustness since, firstly, the secondary MissionPlanner is able to work stand-alone in combination with the central telemetry server and, secondly, the OpenMCT-server has been outfitted with new display features capable of displaying the same telemetry data in the same way, with some additions like further displays. The following Figure 23 shows the map planned to be used for the flight test campaign.
4. Addition of data displays facilitating data evaluation during test flights: Further displays were added to the range of available plugins in the OpenMCT software. These comprise among others a *V-n*-diagram, displaying the current aircraft state and giving an overview over the attained aircraft states during push-over-pull-up maneuvers. A sample is given in Figure 24. Another feature worth mentioning is the implementation of a rolling data record, that will facilitate the saving of telemetry data as well as their reply, which is expected to greatly increase the efficiency of flight test efficiency.
5. Increased inter-connectivity and partner involvement: The Ground Control Station van is planned to be outfitted with a LTE-router that can serve as an internet gateway. Using a VPN-tunnel, the local Open MCT server communicates with a mirrored instance hosted at the LRZ-cloud of TUM. Using personalized keys, partners are planned to be able to login and view the data during flight tests. Initial tests within the eduroam-network of TUM qualitatively compared the latency between the data as displayed by the central telemetry server and the data as displayed by the OpenMCT server hosted at LRZ without finding substantial lags. Lags may however be introduced by the network quality in between the servers (LTE-network) and client computers. Nevertheless, it is expected that this feature will give increased opportunity to partners to get involved and sanity check data while being in-field.

Telemetry (SZTAKI, TUM)

The telemetry system is still a crucial part of ground control station. That is why the development teams at TUM and SZTAKI keep developing the telemetry connection between the T-Flex and the GCS van to be more reliable, fast, and to expand the amount of data sent.

The dev team at TUM has been experimenting with the ground control station visualisation software from NASA called OpenMCT. OpenMCT is a web-based platform available for desktop and mobile as well. We would use it to replace the EDL messaging system, because OpenMCT provides more flexible

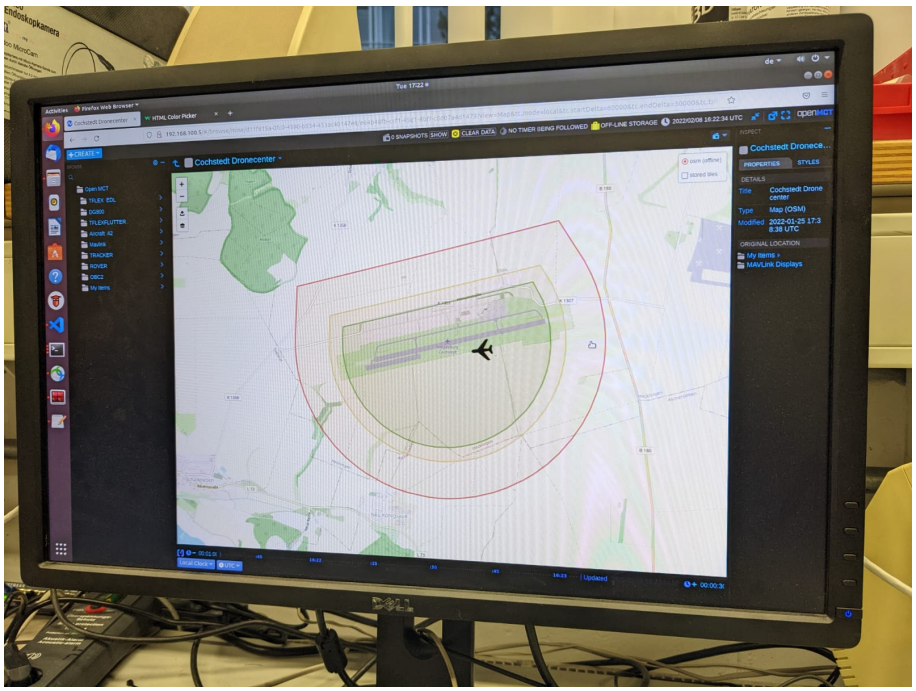


Figure 23: The moving map planned to be used for the flight test campaign at EDBC. The map contains the flight box (shaded in green), the contingency box (shaded in orange) and the perimeter (shaded in red).

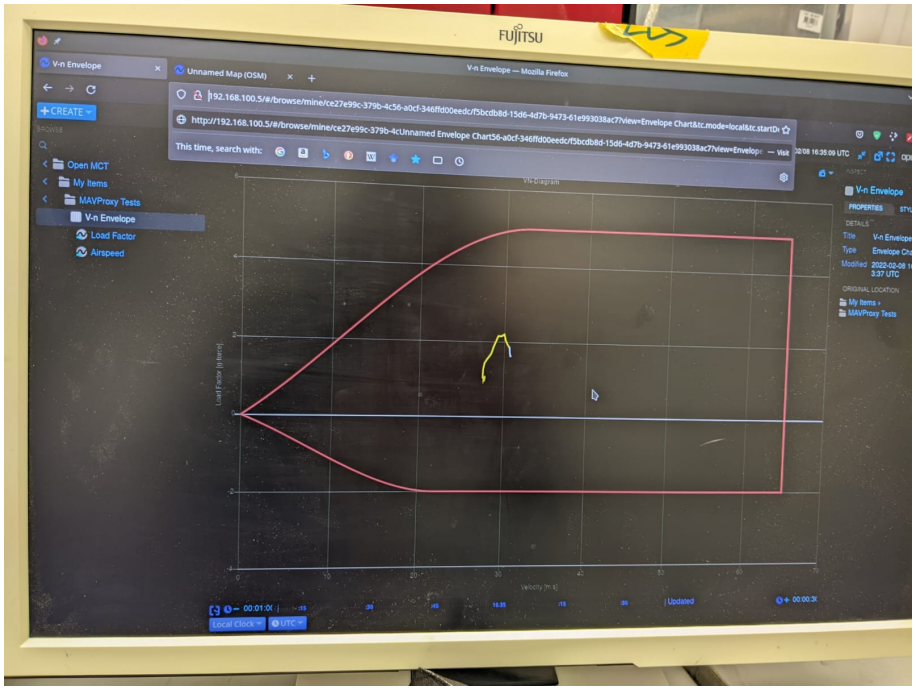


Figure 24: *V*-*n*-diagram

and quicker development of data visualisation interfaces and it does not require MATLAB, just a web browser, preferably Chrome. This new platform still uses the 433 MHz radios to establish physical connection. OpenMCT running on one of the GCS laptops can be seen in Figure 25. The other

Figure 25: OpenMCT running on a GCS laptop

communication system in use, which relies on MAVlink stayed basically the same, but there are going to be new features in the custom-made autopilot interface. Namely, there are going be new parameters to tune the behaviour of the autothrottle controller. The Mission Planner interface is ready, only the parameters have to be defined.

There is a third telemetry link planned to be integrated into the aircraft. This would be connected to the newly developed and installed OBC-II (On-Board Computer 2) which is designed to run the on-board modal analysis software (described in section 3.5) and to act as a host to additional sensors like the TMS (Thrust Measurement Sensor). So far, the same telemtry radios are used as in the old two systems, but TUM plans to upgrade it to a high bandwidth wifi solution. The GCS interface for this is the aforementioned OpenMCT, which visualises the output of the modal analysis software in real time. For an overview, the current telemetry setup of the T-Flex can be seen in Figure 26.

3.5 Operational Modal Analysis (DLR-AE)

Operational Modal Analysis (OMA) will be performed in real time 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. Furthermore, the system will identify and log the natural frequencies, damping ratios and mode shapes of the structure using a time domain method called Stochastic Subspace Identification (SSI) and a frequency domain method called Least Squares Complex Frequency (LSCF). As the system matures during testing, a connection to the controller providing real time state matrices could be further investigated.

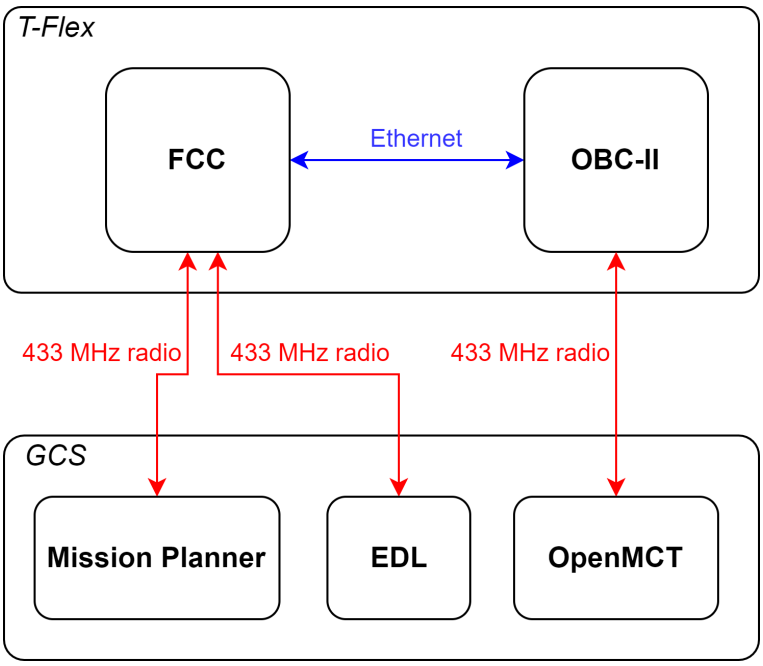


Figure 26: Latest telemetry setup of the T-Flex demonstrator

During flight testing the system should identify and track the modal parameters during normal flight conditions, i.e. without additional excitation. Depending on the amplitude of the excitation forces, the duration of the time data, or the amount of time spent on a stationary flight point will vary in order to achieve a given uncertainty. If longer duration's of data are required than can be achieved in the flight box, repetition of the horse race pattern could be used.

The data will be transmitted using 433 MHz telemetry and piped through a Python-InfluxDB-OpenMCT pipeline. The tracked natural frequencies and damping ratios will be displayed on a monitor in real time.

3.6 Documentation (TUM)

The flight test documentation takes place in three areas: logged data, video data and written notes.

As before, the flight tests will be logged on the FCC. Additionally, thrust measurement data will be logged on the OBC-II platform.

Videos will be recorded on the two tail-mounted cameras. The GCS screens will be recorded for backup. Additionally, already for the last two flights in 2021, a 360 degree camera has been mounted above the fuselage to have a better overview of the aircraft in-flight. An example of the panoramic view from the camera is shown in Figure 27. Notes will be made on the checklists and test cards. For the past two flights TUM has been testing out a way to make the test cards digitally by using a tablet. It seemed to work well and was especially helpful taking more notes during the debriefing. Furthermore, as the notes don't have to be digitized, they are immediately uploaded together with the test data.

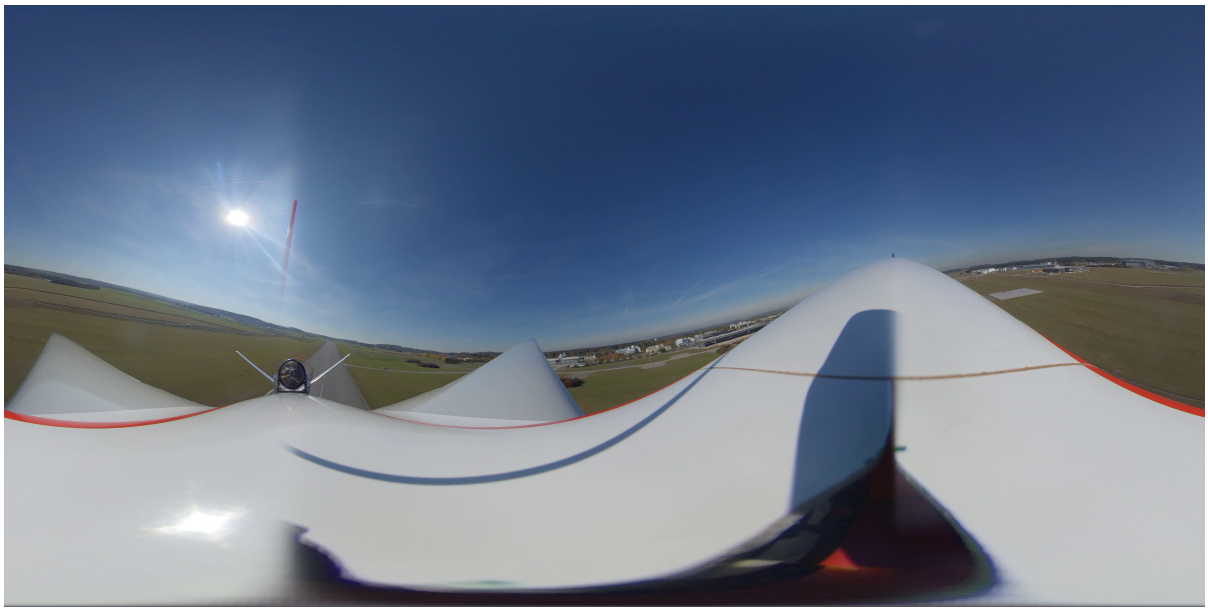


Figure 27: Panoramic view from the 360 degree camera mounted above the fuselage. Visible is (from left to right): left wing, engine and tail, right wing, front-top part of the fuselage.

4 Flight Test Programme

At the moment of writing, only three test flights have been performed within FliPASED. The reasons for this are already explained within the D3.1 Flight Test Report Phase 1.

Graphically, the flight test progress is represented in Figure 28.

4.1 Flight Tests with the -1 wing (all)

4.1.1 Maiden Flight (TUM)

The maiden flight of -1 wing will be done with the flutter actuator already mounted on the wing. However, as a precaution, no flutter masses will be mounted for the initial flights, or until the flutter controllers will be made functional. In this case the flutter speeds will remain high enough to perform the usual flight performance tests before approaching the dangerous part of the flight configuration envelope.

During the first flight, the aircraft correct functioning of the -1 wing control systems will be checked. The trim will be done and full functionality of the flutter actuator will be checked.

4.1.2 Rigid Mode Identification (TUM)

The flutter masses will be installed and flutter configuration will be flown, taking great care not to approach the flutter speeds. Envelope will be expanded to a safe margin below the flutter speed during the second flight.

During this test phase, traditional or orthogonal multi-sine input manoeuvres will be injected to trigger the rigid body modes. More experience on these manoeuvres will be collected during the first part of the flight testing in 2022. There, similar manoeuvres will be applied to the -0 wing in order to test the efficiency of the aerodynamic system identification methods.

4.1.3 Flexible Mode Identification (TUM)

Similar as before, different manoeuvres will be applied to identify and correct the flexible modes of the flying aircraft.

After rigid body modes and aeroelastic modes are identified, a review of the data will take place, during which the model used fro flutter controller will be updated based on flight test data. When the applicability of models will be confirmed, only then it will be safe to start the flutter tests.

4.1.4 Flutter Test (DLR-SR,SZTAKI integrate information to a minimum number of flight tests)

After the previous tests have been successfully conducted the flutter tests can take place. Three different kinds of flutter tests are considered. They are sorted by increasing challenge. For all the flight controller is switched on, in order to fly the horserace pattern and operate the aircraft within a very tight speed range, so the flutter speed can not be suddenly exceeded. In the following the flutter tests are further described.

Flutter Test 1:

flight controller:	on (flying horserace pattern)
flight speed:	below open-loop flutter speed with stepwise increase (30-$40\,\mathrm{m/s}$)
flutter controller:	off
flutter tuning masses:	detached
operational modal analysis:	active

Flutter Test 2:

flight controller:	on (flying horserace pattern)
flight speed:	below open-loop flutter speed with stepwise increase (35-45 m/s)
flutter controller:	on
flutter tuning masses:	attached
operational modal analysis:	active

Flutter Test 3:

flight controller:	on (flying horserace pattern)
flight speed:	beyond open-loop flutter speed with stepwise increase (45-55 m/s)
flutter controller:	on
flutter tuning masses:	attached
operational modal analysis:	active

4.2 Post Flight Procedure (all)

4.2.1 Procedure from TUM (TUM)

Immediately after the test flight, the data is transferred to one of the data processing laptops. Data processing scripts are launched, which prepare the data for further processing.

Additionally, the automatically generated Preliminary Flight Test Report is evaluated (Figure 29). The report displays the flight trajectory, flight phases and, most importantly, shows if any abnormal conditions were triggered, such as high servo temperatures or load factors. At this stage, the manoeuvres as required by the partners can be extracted and trimmed using the Evaluation Tool (Figure 30) which is currently in development. The tool has a graphical user interface which shows the trajectory of the selected segment and can replay a 3D animation of the segment. It also plots the selected parameters to help the engineer identify if the segment will fit the requirements of analysis. Comments can then be added and the segment extracted for further analysis.

4.2.2 procedure from DLR-AE (DLR-AE)

After a flight test certain time windows are extracted from the measurements to analyze the mode shapes including their frequency and damping. The time windows should be at least 30 seconds long and exhibit a sufficient excitation of the wing IMUs. Moreover, the flight condition should stay roughly the same within a window. To gain a better understanding of the aircraft, windows featuring different flight conditions can be compared. Thus, changes in the mode shapes due to, e.g., different velocities are observable. The results can then be provided to the partners for comparison.

4.2.3 procedure from DLR-SR (DLR-SR)

The mode shapes, frequencies and dampings obtained from the operational modal analysis can be compared with the model provided for the synthesis of the flight controller and the flutter controller. In case the match is within a predefined bound of uncertainties, the selected models are accurate enough. If this is not the case the model is adapted based on the flight test data and the results of the operational modal analysis. Subsequently, the flight and flutter controllers need to be retuned.

4.2.4 procedure of SZTAKI (SZTAKI)

The damping of flexible modes is evaluated after the flight test. It is expected that the flutter controller adds damping to the flexible modes even below the flutter speed. This is a crucial test since it indicates the effectiveness of the flutter controller without taking the risk of flying above flutter speed. If the damping is increased as expected then the next flights can be conducted. If not, it might be because

the model needs to be updated. After the model is updated, the flutter controller needs to be retuned and the first flight test is repeated.

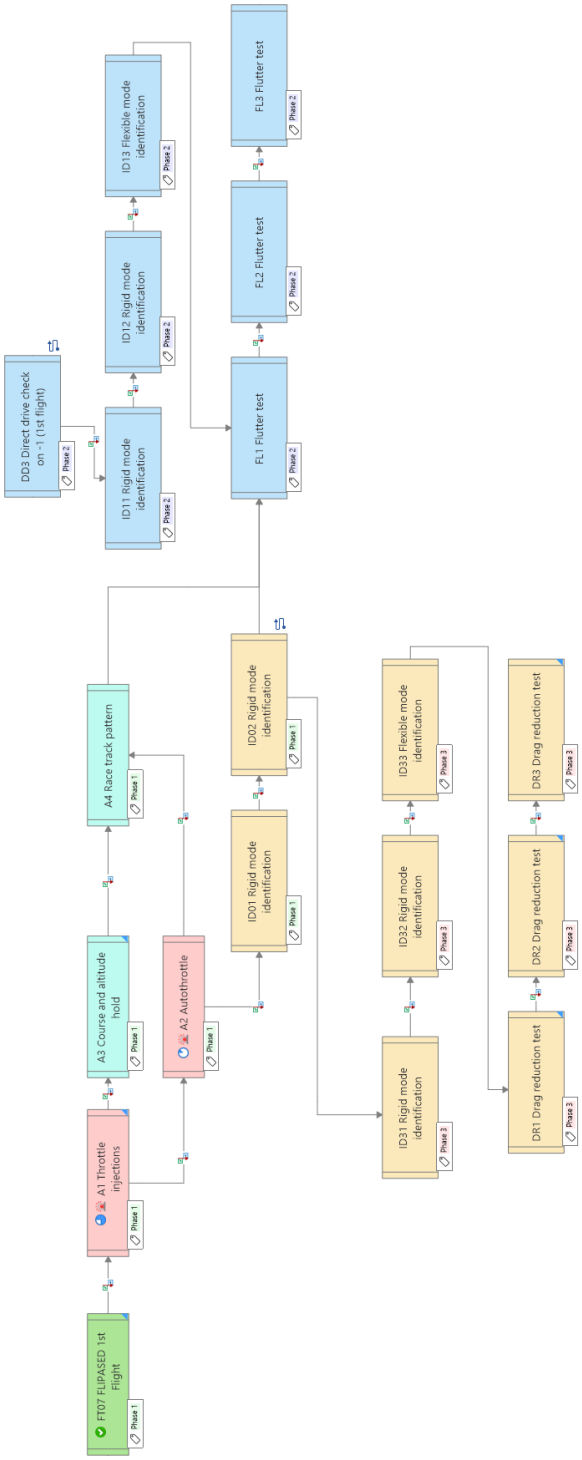


Figure 28: Flight test phases within FliPASED. Here phase colours are: green- completed, red- late, turquoise- autopilot functionality testing, yellow- -0 and -3 wing testing phases, blue- -1 wing testing phases.

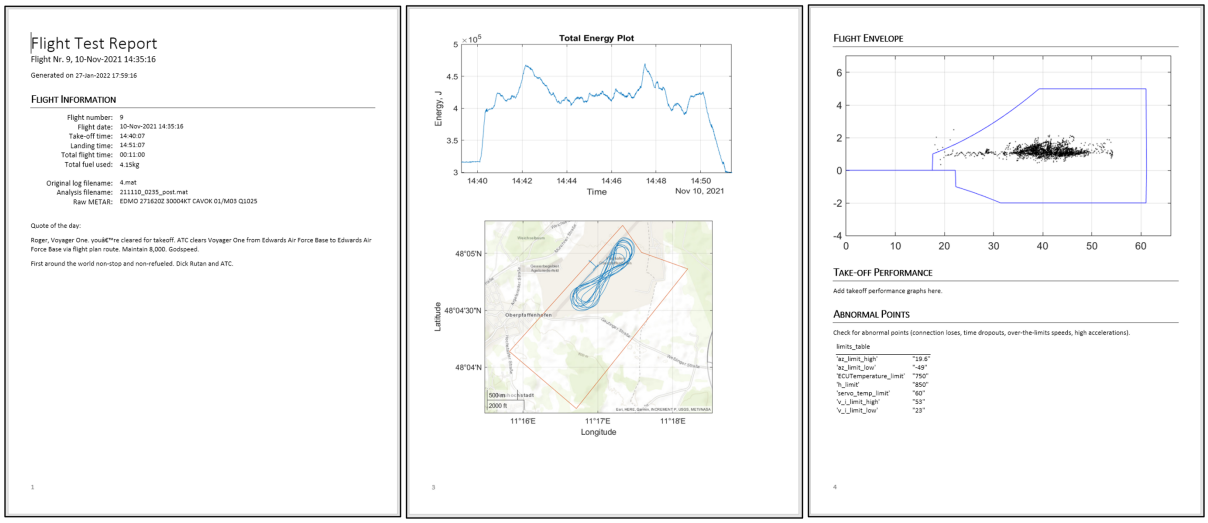


Figure 29: The automatically generated Preliminary Flight Test Report. It displays the flight trajectory, main UAV parameters and if any of the pre-programmed triggers were set.

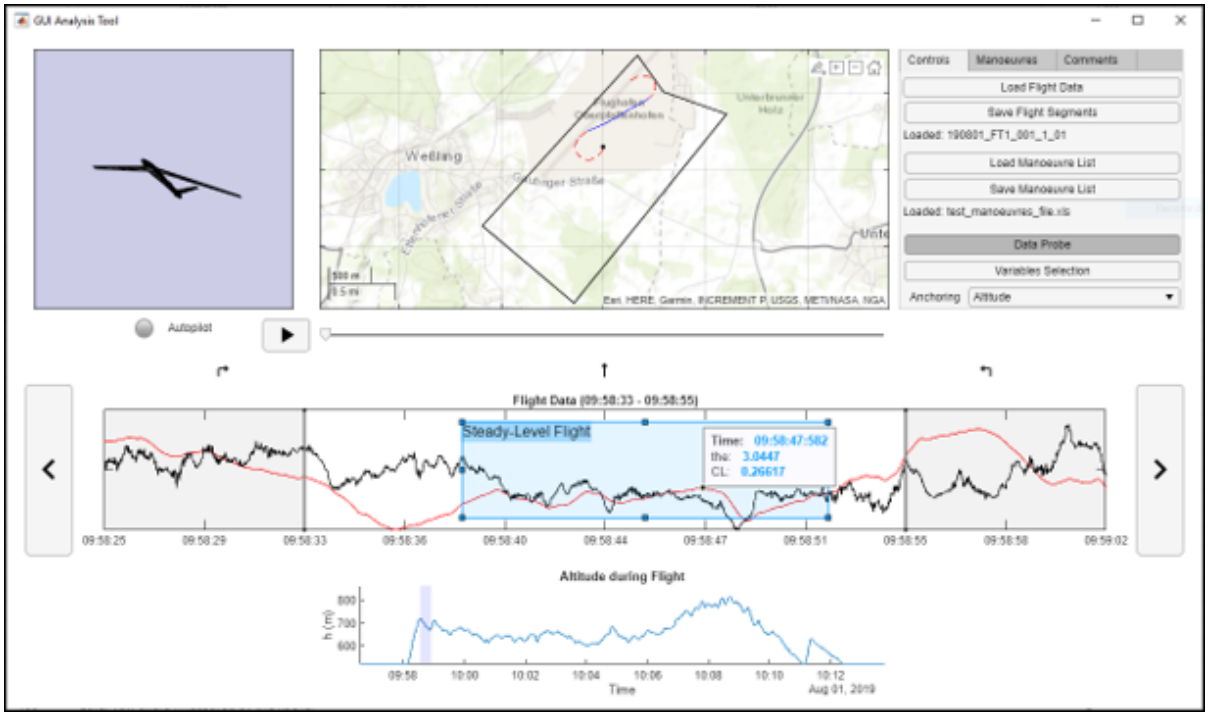


Figure 30: The GUI of the Flight Test Data Evaluation Tool. The screen shows the trajectory of the selected segment, 3D animation of the UAV and plots the selected parameters for the segment. Comments can then be added and the segment extracted for further analysis.

5 Conclusion

The second flight test phase of the demonstrator aircraft with the -1 wing pair attached has been planned. The flight tests are incrementally developed for a save handling of a flutter-critical aircraft. Each flight test will be analysed in detail in order to prove or correct the beforehand predictions. The final goal of the flight test campaign is to fly the demonstrator above the open-loop flutter speed with an active flutter suppression stabilizing the aircraft. As a cautious approach has to be taken along the way to active flutter suppression, this test plan is meant to be a living document that is iterated with respect to new requirements.

6 Bibliography

[1] Andreas Hermanutz. Flexop D1.4 report on design of -1 wings. Technical report, TUM, 2017.

[2] Manuel Pusch. *Blending of Inputs and Outputs for Modal Control of Aeroelastic Systems*. PhD thesis, Technical University of Hamburg, 2020.

[3] Keith Soal, Richard Kuchar, Julius Bartasevicius, Charles Poussot-Vassal, and Szabolcs Toth. Flipased D1.3 demonstrator ground and flight test requirements definition. Technical report, SZTAKI, 2019.

[4] 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. In *AIAA Scitech 2020 Forum*.