D4.10 Release of the flight test results and models of the a/c for the community
Balint Vanek (SZTAKI)

GA number: 636307
Project acronym: FLEXOP
Project title: FLUTTER FREE FLIGHT ENVELOPE EXPANSION FOR ECONOMICAL PERFORMANCE IMPROVEMENT

Funding Scheme: H2020 MG-1.1-2014
Latest version of Annex I: 2.1 released on 20/11/2015
Start date of project: 01/06/2015 Duration: 54 Months

<table>
<thead>
<tr>
<th>Lead Beneficiary for this deliverable:</th>
<th>SZTAKI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last modified: 30/11/2019</td>
<td>Status: Delivered</td>
</tr>
<tr>
<td>Due date: 30/11/2019</td>
<td></td>
</tr>
</tbody>
</table>

Project coordinator name and organisation: Bálint Vanek, SZTAKI
Tel: +36 1 279 6113
Fax: +36 1 466 7483
E-mail: vanek@sztaki.mta.hu
Project website: www.flexop.eu

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td>Public</td>
</tr>
<tr>
<td>CO</td>
<td>Confidential, only for members of the consortium (including the Commission Services)</td>
</tr>
</tbody>
</table>

“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 636307.”
Table of Contents

Table of Contents ........................................................................................................................................ 2

1 Objectives of this report .................................................................................................................. 4

2 Aeroservoelastic Modelling ........................................................................................................... 6
   2.1 Structural Dynamics ............................................................................................................... 6
   2.1.1 Condensed Model ........................................................................................................... 6
   2.1.2 Equations of Motion ....................................................................................................... 6
   2.1.3 Rigid Body Motion ......................................................................................................... 7
   2.1.4 Elastic Body Motion ....................................................................................................... 7
   2.2 Aerodynamics ....................................................................................................................... 7
   2.2.1 Panel Model ................................................................................................................... 7
   2.2.2 Steady Aerodynamics via VLM ...................................................................................... 8
   2.2.3 Unsteady Aerodynamics via DLM ................................................................................... 9

2.3 Model Integration .................................................................................................................... 10

3 Model Order Reduction Techniques .......................................................................................... 12
   3.1 Physical based methods ......................................................................................................... 12
       3.1.1 Guyan Reduction ......................................................................................................... 12
       3.1.2 Truncation and Selection of Flexible Modes ............................................................... 13
   3.2 Modal decomposition based methods ................................................................................... 15
       3.2.1 Introduction ................................................................................................................ 15
       3.2.2 Model description ........................................................................................................ 16
       3.2.3 Equations of Motion ................................................................................................... 17
       3.2.4 Aerodynamic forces .................................................................................................. 18
       3.2.5 Model summary ........................................................................................................... 19

3.3 Coordinate definitions and reference values ............................................................................. 22

4 Matlab/Simulink Implementation ................................................................................................. 25
   4.1 Full order model ................................................................................................................... 25
   4.2 Reduced order model ............................................................................................................ 30

5 Onboard Sensors ........................................................................................................................... 32
   5.1 The flutter IMUs (flutterIMU) ............................................................................................... 33
   5.2 The servo health monitoring unit (SHMU) ........................................................................... 35
   5.3 xSens ................................................................................................................................... 37
   5.4 Micro Air Data System ......................................................................................................... 38
   5.5 Shape Monitoring System .................................................................................................... 39
       5.5.1 Monitoring system operation ....................................................................................... 41
5.5.2 General overview of the sensing system layout ......................................................... 42
6 Description of the Flight Test ...................................................................................... 47
7 Flight Test Data ........................................................................................................ 51
    7.1 Log-plotter: Overview ..................................................................................... 51
    7.2 Data Analysis ..................................................................................................... 53
8 Deviations and Conclusion ........................................................................................ 63
1 Objectives of this report

Summary

In accordance with the open research data initiative, flight test results and models of the demonstrator are released to the community. The models and data are documented in this report.

The specific aircraft model, the FLEXOP demonstrator, has the unique characteristics of having interchangeable wings. Hence, data of two different high-fidelity models are included. The model structure is the same for them, but the model parameters are different in the two models included. The -0 configuration is the baseline aircraft, with conventional CFRP wing structure, most of the flight tests have been performed with this wing. The second parameter set corresponds to the -1 configuration, which is significantly more flexible, exhibiting unstable flutter characteristics starting at 50 m/s. The teams of DLR, TUM and SZTAKI have also analysed the flight test data from -0 flights and assessed its quality, what shows good correlation with the model predictions.

A lower fidelity, reduced order version of the -1 aircraft simulation model is also included in the software package.

The models are accompanied by a package of flight test data of the aircraft, conducted with the -0 wings.

Data handling and sharing principles have been detailed in D5.3, Data Management Plan. Sharing of results are handled according to the principles of this report. Open Access data is placed into the Open Access Repository which will be handled by the Coordinator. Its content will also be available for 5 more years after the conclusion of FLEXOP.

The mathematic modelling background is described first in the report (Chapter 2). The Matlab/Simulink implementation of the high fidelity and reduced order modelllas are discussed in Chapter 4, followed by the description of the physical parameters of the demonstrator, including reference locations and dimensions. The descriptions of the sensor systems are discussed in Chapter 5, while the flight test specific flight test card is in Chapter 6. Details and examples from the flight are shown in Chapter 7. The document is concluded in Chapter 8.
The data repository link is available from the project webpage: https://flexop.eu/, direct link to the flight test data, where a password protected zip file can be download is located at: https://flexop.eu/static/misc/flight_test_data.zip

Users interested in obtaining an access code to this archive folder should send an email to vanek@sztaki.hu, to keep track of the users who might want to use the data.

The present document describing the usage of the data is available also on: https://flexop.eu/static/misc/D410_FLEXOP_y2019m11d30.pdf
2 Aeroservoelastic Modelling

2.1 Structural Dynamics

The structural dynamics of the flexible aircraft can be divided into a rigid-body and an elastic-body motion. The rigid-body motion basically describes the maneuver characteristics of the aircraft. In contrast, the elastic-body motion represents the dynamics due to the flexibility of the aircraft structure. While the rigid-body motion is described in nonlinear form, the equation of the elastic-body motion is considered to be linear. A detailed finite element (FE) model serves as basis for the structural model of the aircraft. The process of generating the FE model and its condensed version is described below. Subsequently the EOM representing the rigid and elastic-body motion are defined for the condensed model (1).

2.1.1 Condensed Model

The aircraft structural FE model comprises the wing, fuselage and empennage and is shown in Figure 2.1. The FE software used here is MSC.NASTRAN.

![Figure 2.1: Full FE model of the FLEXOP demonstrator aircraft](image)

Given that the FE model of the wing is of very high-fidelity (more than 600000 nodes), a Guyan-reduction, also called condensation, is performed reducing the mass and stiffness matrix between 100 and 200 nodes in the condensed model (2).

2.1.2 Equations of Motion

The condensed model features rigid body and flexible modes, which are described by the EOM. The EOM are based on an equilibrium of forces and moments. They describe the behavior of the aircraft due to external loads originating from the aerodynamics and thrust. For simplification, the following assumptions are made.

As the earth rotation can be neglected, the inertial reference system is earth fixed.
Gravity is constant over the airframe,
The deformations of the airframe are considered to be small which allows the use of linear elastic theory defined by Hooke’s law,
Due to small deformations of the aircraft structure, the aircraft mass moment of inertia \( J_0 \) remains unchanged,
As the structural deformations are small, loads act on the undeformed airframe,
The eigenvectors of the modal analysis are orthogonal, because of which the total structural deformation can be written as a linear combination of the modal deflections,
The rigid body and flexible body EOM are considered to be decoupled.

2.1.3 Rigid Body Motion

For the derivation of the nonlinear flight mechanical EOM, the aircraft is considered as a rigid body with a constant mass $m_b$ and constant mass moment of inertia $J_b$. Therefore, the aircraft rigid-body motion is described by

$$
\begin{bmatrix}
    m_b (v_b + \Omega_b \times v_b - T_{pe} g_e) \\
    J_b \Omega_b + \Omega_b \times (J_b \Omega_b)
\end{bmatrix} = \Phi_{gb}^{T} p_{ext}^{gb}(t),
$$

which represents the Newton-Euler EOM. In Equation 1 the translational and angular velocity of the aircraft with respect to the body frame of reference are given by $v_b$ and $\Omega_b$. The vector $g_e$ represents the gravitational acceleration, which is transformed with $T_{be}$ from the earth fixed to the body fixed frame of reference. The external loads $p_{ext}^{gb}(t)$ acting on the aircraft structure are transformed by the transpose of $\Phi_{gb}^{T}$ into the rigid body frame (1).

2.1.4 Elastic Body Motion

As the displacements due to the aircraft flexibility are assumed to be small, linear elastic theory is applied to define the elastic motion. Therefore, the correlation between external loads $p_{ext}^{gf}(t)$ and the generalized coordinates $u_f$ representing the modal deformation of the structure is given by the differential equation

$$
M_{ff} \dddot{u}_f + B_{ff} \ddot{u}_f + K_{ff} u_f = \Phi_{gf}^{T} p_{ext}^{gf}(t),
$$

where $M_{ff}$, $B_{ff}$, and $K_{ff}$ are the modal mass, damping and stiffness matrices. The modal matrix $\Phi_{gf}$ contains the eigenvectors of the structural modes sorted by frequency (1). Typically, higher frequencies have a smaller contribution to the overall system performance. Consequently, modal truncation can be applied to reduce the DOF significantly by considering only the most relevant eigenmodes. Thus, modal truncation is part of the model order reduction and will be described in more detail later.

2.2 Aerodynamics

The aerodynamic loads represent the major external loads acting on the aircraft structure. Their calculation is based on the VLM for steady aerodynamics and the DLM for unsteady aerodynamics. Both methods are based on a panel model, which is described in the following section.

2.2.1 Panel Model

The lifting surfaces are discretized by several trapezoidal shaped panels, known as aerodynamic boxes as shown in Figure 2.2. Of note is the panel model for the fuselage. The wetted areas of the fuselage are projected onto a T-cruciform shaped panel model. Although this is a vast simplification, the fuselage aerodynamics are modeled quite accurately with respect to higher-order computational fluid dynamics (CFD) simulations.
2.2.2 Steady Aerodynamics via VLM

The VLM is used to model steady aerodynamics. As can be seen in Figure 2.3 (left), each aerodynamic box of the panel model possesses a horseshoe vortex at point $l$ on the quarter-chord line.

![Figure 2.2: Aerodynamic boxes of the FLEXOP demonstrator aircraft](image)

![Figure 2.3: Aerodynamic box top (left) and side view (right)](image)

Due to the Helmholtz theorem the vortex is shed downstream to infinity at the side edges of the box. For each aerodynamic box the Pistolesi Theorem needs to be met, stating that there is no perpendicular flow through the control point $j$ at the three-quarter-chord line. Therefore the induced velocity at the control point needs to equalize the perpendicular component of the incoming flow, like shown in Figure 2.3 (right). By means of the Biot-Savart law the induced velocities $v_j$ due to the circulation strengths $\Gamma_j$ of the horseshoe vortices can be determined by

$$v_j = A_{jj} \Gamma_j$$

The matrix $A_{jj}$ describes the contribution of all vortices to the induced velocities of the aerodynamic boxes. Inverting $A_{jj}$ and multiplying with $2/c_T$ where $c_j$ is the chord length of the respective aerodynamic box, leads to the aerodynamic influence coefficient (AIC) matrix $Q_{jj}$. It is used to determine the pressure coefficient.
where $w_j = \frac{v_j}{u_{\infty}}$ is the downwash $v_j$ normalized with the flight speed $u_{\infty}$. It is assumed to be equal to the angle of attack $\alpha$, i.e. $w_j = \sin(\alpha) \approx \alpha$, as only small angles are considered.

There are different contributions, that determine the value of $w_j$. The VLM is based on lift coefficient gradients, meaning the change in lift can be predicted very well. In order to receive the absolute lift coefficients, they have to be initialized by a reference calculation. Therefore, the downwash has to be corrected by means of $w_{j,0}$. Within the FLEXOP project a steady CFD calculation at an angle of attack $\alpha = 0$ is used. Rigid-body motions of the aircraft also affect the downwash. For example, a downward motion of the aircraft changes the direction of the incoming flow and therefore the angle of attack. $w_{j,b}$ accounts for the contribution due to a rigid-body motion. Also the deflection of the control surfaces affects the steady aerodynamics. By deflecting the aerodynamic boxes, belonging to the control surfaces, the downwash is changed by $w_{j,c}$. Not only the deflection of the control surfaces alters the lift, but also their deflection rate, which can be accounted for by $w_{j,\dot{c}}$.

So far only aerodynamic effects of a rigid-body have been considered. In order to consider aeroelastic effects, the downwash has to be dependent on flexible motions. Comparable to the control surfaces, the flexible deflection and deflection rate of the aircraft contribute with $w_{j,\delta}$ and $w_{j,\dot{\delta}}$ to the downwash (1; 4; 5; 6). Finally the equation for the downwash is given by

$$w_j = w_{j,0} + w_{j,b} + w_{j,c} + w_{j,\dot{c}} + w_{j,\delta} + w_{j,\dot{\delta}}.$$

### 2.2.3 Unsteady Aerodynamics via DLM

In order to take unsteady aerodynamic effects into account, the aerodynamic model has to be widened. When an airfoil is suddenly moved forward at an angle of attack, it creates circulation. Since the Helmholtz theorem states that the total circulation has to stay constant, a vortex of the same strength but with opposite direction of rotation has to be shed from the trailing edge. The vortex moves downstream losing its influence on the airfoil the farther away it gets. Its effect therefore decreases with time and the flow converges to the steady condition. This lagging effect is caused by unsteady aerodynamics (4).

Unsteady aerodynamics are covered by the Doublet Lattice Method (DLM). Instead of horseshoe vortices, doublets are placed at the quarter-chord line of each aerodynamic box. In contrast to the VLM, the pressure coefficient is determined by

$$\Delta c_{p,j}(k) = q_{jj}(k)w_j(k)$$

in the reduced frequency domain, where the dimensionless reduced frequency is

$$k = \frac{\omega c_r/2}{u_{\infty}}.$$

In Equation 7, $c_r$ depicts the reference chord length and $\omega$ is the frequency. For $k = 0$ the quasi-steady solution is derived. In order to be able to transform the unsteady aerodynamics to the time domain, a rational function approximation (RFA) is chosen. The AIC matrix is approximated with Roger’s method, which is described in detail in Ref. (7), as...
\[ Q_{jj}(s^*) = Q_{0,ij} + Q_{ij} s^* + \sum_{i=1}^{\eta_p} Q_{li,ij} \frac{s^* I}{s^* + p_i} \]

where \( s^* = ik \) is equivalent to the Laplace variable \( s \) for the reduced frequency \( k \). When multiplying with the downwash \( w_j \) from the right side and performing an inverse Laplace transformation, Equation 8 becomes

\[ \Delta c_{ij} = L^{-1}(Q_{ij}(s^*)w_j(s^*)) \]

\[ = \frac{Q_{0,ij}w_j}{\text{quasi-steady}} + \left( Q_{ij} + \left[ Q_{il,ij} \ldots Q_{lp,ij} \right] L^{-1}\left( \sum_{i=1}^{\eta_p} \frac{Q_{li,ij} I}{s^* + p_i} \right) \right) \frac{U_{ma}}{U_{ma}} w_j \]

The unsteady aerodynamics can be represented as a state space system, defined by

\[ \dot{x}_L = Ax_L + Bw_j \]

\[ \Delta c_{ij,\text{unsteady}} = Cx_L + Dw_j \]

\[ A = \text{diag}(-p_2 I \ldots -p_{\eta_p} I) \left( \frac{U_{ma}}{U_{ma}^2} \right) \]

\[ B = [I \ldots I]^T \]

\[ C = [Q_{1,ij} \ldots Q_{lp,ij}] \]

\[ D = Q_{ij} \left( \frac{U_{ma}}{U_{ma}} \right) \]

The new states \( x_L \) are called lag states. They represent the lagging behavior, which is caused by unsteady aerodynamics.

It can be seen, that the first term in Equation 9 depends on the downwash \( w_j \), while the second term depends on its derivative \( \dot{w}_j \). Therefore, it can be clearly distinguished between quasi-steady and unsteady aerodynamics. By differentiating Equation 5 \( \dot{w}_j \) is obtained by

\[ \dot{w}_j = \dot{w}_{j_{bo}} + \dot{w}_{j_{bo}} + \dot{w}_{j_{x_0}} + \dot{w}_{j_{x_1}} + \dot{w}_{j_{f_0}} + \dot{w}_{j_{f_1}} \]

where \( w_{j_{bo}} \) is considered to be constant and therefore its derivative \( \dot{w}_{j_{bo}} \) is equal to zero. The rigid-body accelerations, the control surface deflection rates and accelerations as well as the flexible-body deflection rates and accelerations cause unsteady aerodynamic effects and therefore a change in the downwash (1).

### 2.3 Model Integration

The overall aeroservoelastic model is built of several sub-models, as depicted in Figure 2.4.
Figure 2.4: Aeroservoelastic system
The core forms the aeroelastic system, which represents the coupling of aerodynamics and structural dynamics. The aerodynamic loads $p_{aero}^{g}$ directly correlate with the structural motion and can be expressed in terms of $\Delta c_{p_{j}}$ as

$$p_{aero}^{g} = q_{\infty} T_{bg}^{T} S_{kj} \Delta c_{p_{j}},$$

where $S_{kj}$ is an integration matrix relating the pressure in the aerodynamic boxes at point $j$ with the forces at the aerodynamic grid points $k$. The forces at the aerodynamic grid points $k$ are then interpolated onto the structural grid points via the transpose of the spline matrix $T_{bg}$. The splining model for the wing is shown as an example in Figure 2.5.

Figure 2.5: Splining between the aerodynamic model and structural model of the right wing

The structural nodes used for the splining comprise of the condensation nodes as well as two nodes per condensation node, extended to the leading and trailing edge at that point using rigid elements. A similar splining model is used for the empennage as well. Multiplying with the dynamic pressure $q_{\infty}$ leads then to the aerodynamic loads acting on the structure. The aircraft structure reacts on the aerodynamic loads by performing rigid-body and flexible motions, which directly affect the aircraft aerodynamics. Therefore the aeroelastic model is considered as a loop between structural dynamics and aerodynamics. An aeroservoelastic model is derived by adding an active flutter control system to the aeroelastic model. The active flutter control system comprises of actuators for deflecting the relevant control surfaces, sensors for observing the dynamic behavior of the aircraft and the flutter controller.
3 Model Order Reduction Techniques

3.1 Physical based methods

The physical based model order reduction techniques presented in this section aim on a reduction of the DOF of the structural model. This also leads to a reduction of the aeroelastic model.

3.1.1 Guyan Reduction

The first step of the model order reduction is the Guyan reduction. Like mentioned in previously the Guyan reduction allows reducing the DOF of the aircraft FE model. It basically neglects DOF, however the stiffness matrix $K$ and the mass matrix $M$ of the aircraft need to be recalculated. The static equation of a FE model is given by

$$ P_{\text{ext}} = Kx, $$

Where $P_{\text{ext}}$ are the loads applied to the aircraft structure. Equation 18 can be rewritten as follows:

$$ \begin{bmatrix} P_{\text{ext,1}} \\ P_{\text{ext,2}} \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} $$

It is separated in DOF $x_1$, which are meant to keep, and the DOF $x_2$, which to neglect. Therefore, the loads $P_{\text{ext,2}}$ are considered to be equal to zero. Equation 19 therefore becomes

$$ P_{\text{ext,1}} = (K_{11} - K_{12}K_{22}^{-1}K_{12}^T)x_1. $$

The reduced stiffness matrix is considered to be

$$ K_1 = K_{11} - K_{12}K_{22}^{-1}K_{12}^T. $$

The Guyan reduction can be considered to be a transformation of the form

$$ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} I \\ -K_{22}^{-1}K_{12}^T \end{bmatrix} \begin{bmatrix} x_1 \end{bmatrix}. $$

Matrix $\tau$ depicts the transformation matrix. The structure energies are defined by

$$ E_M = \frac{1}{2} \dot{x}_1^T T^T MT \dot{x}_1, $$

$$ E_K = \frac{1}{2} \dot{x}_1^T T^T KT \dot{x}_1, $$

Where $M_2 = T^T MT$ is considered to be the reduced mass matrix. With

$$ M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} $$

The reduced mass matrix can be determined to be

$$ M_1 = M_{11} - M_{12}K_{22}^{-1}K_{12}^T - (K_{22}^{-1}K_{12}^T)^T (M_{22} - M_{22}K_{22}^{-1}K_{12}^T). $$
It can be seen, that the reduced mass matrix $M_1$ is combined of the original mass and stiffness matrices. Thus, the system is not exactly the same as the original problem. However, the number of DOF can be significantly. It needs to be examined, if the relevant dynamics are preserved (2).

### 3.1.2 Truncation and Selection of Flexible Modes

The number of states of the aeroelastic system is defined by

$$n = 2n_b + 2n_f + n_{lag},$$

where $n_b$ is the number of rigid-body modes, which is fixed to 6, $n_f$ is the number of the considered flexible modes and $n_{lag}$ is the number of lag states. The number of lag states is dependent on the kind of realization. Here Equation 9 is input-realized, therefore the number of lag states is calculated in terms of

$$n_{lag} = n_{poles}(n_b + 2n_{x2} + 2n_f).$$

In Equation 28, the number of control surfaces $n_{x2}$ is fixed to 12. The number of poles $n_{poles}$ for the RFA is pre-defined and also does not change. In this section the considered model order reduction methods focus on optimizing the number of flexible states $n_f$, i.e. reducing the number of flexible states and still represent the aircraft dynamics accurately.

After the Guyan reduction the order of the aeroelastic model is defined by

$$n = 2n_b + 2n_f + n_{poles}(n_b + 2n_{x2} + 2n_f) = 12 + 1968 + 8 \cdot (6 + 24 + 1968) = 17964.$$

The structural model features as many structural modes as it possesses DOF. Typically, higher frequency modes have a smaller contribution to the overall system performance. Therefore, the amount of the considered structural modes can be limited. The number of modes is set from 984 to 50 flexible modes. The 6 rigid body modes are kept. By truncating higher frequency modes, the number of lag states is reduced as well. The order of the aeroelastic then becomes

$$n = 12 + 100 + 8 \cdot (6 + 24 + 100) = 1152.$$

Still the order can be further reduced. Figure 3.1 shows the modal decomposition of the first 16 flexible modes to symmetric and antisymmetric flutter at 52 $m/s$, which is around the flutter speed.
The contribution of the structural modes from 17 to 50 becomes even less. Therefore, the modes from 17 to 50 can be truncated as well, as they are irrelevant for the flutter phenomena. Mode 12 almost does not contribute to flutter and can be neglected. The aeroelastic system can be rebuilt with 15 instead of 50 flexible modes. This further reduces the number of states to

\[ n = 12 + 30 + 8 \cdot (6 + 24 + 30) = 522. \]

In Figure 3.2 it can be seen, that the modal participation for 15 structural modes has slightly changed. It is therefore essential to examine the performance on a higher order model, after designing a flutter controller.
3.2 Modal decomposition based methods

The present chapter shows the application of control oriented reduced order modeling of a flexible winged aircraft based on approximate modal decomposition.

3.2.1 Introduction

This chapter presents a systematic framework for developing a dynamical model for flexible aircrafts, which is suitable for model-based control design. In order to successfully design a controller, which suppresses these adverse aeroelastic effects, a suitable dynamic model is needed. Aeroservoelastic models can be constructed based on a subsystem approach. First, a linear structural model is generated by finite element method (FEM), rigid body dynamics are replaced by non-linear equations of motion and then it is interconnected with a linear unsteady aerodynamic model generated by Double Lattice Method (DLM). Since the structural damping changes with increasing airflow speed, the model is obtained at different airspeed as a linear, hence the dynamics are in a linear-parameter varying (LPV) form. In order to capture the relevant aeroelastic effects an accurate model is needed, which requires the use of a suitably dense structural grid and large number of lag states in the aerodynamic model. This results in a high-dimensional dynamical system with 522 state variables. Unfortunately this is intractable by the currently available analysis and control synthesis algorithms developed for LPV systems. This makes it necessary to develop an appropriate model order reduction method, which finds a lower dimensional representation for the same dynamical behavior.

The chapter is organized as follows. After the introduction we turn our attention on the development of the model. Section 2 discusses the modeling process to obtain a dynamical model of the flexible winged aircraft. In the following Section 3 we briefly summarize our recently developed LPV model reduction technique. The interested reader is referred to (12; 13) for more details on the algorithm and
to (14) and (15) for application examples. However, the present paper shows the reduction of a more complex model, therefore Section 4 is fully dedicated to the numerical results.

Figure 3.3: Control surface configuration

Figure 3.4: The FLEXOP demonstrator UAV

3.2.2 Model description

The FLEXOP demonstrator UAV is illustrated in Figure 3.3. The design features a wing span of 7 meters at an aspect ratio of 20. The takeoff weight is typically 55 kg but can be increased by up to 11 kg of ballast. The aircraft is equipped with a 300 N jet engine, located on the fuselage back. An air-brake system, deflecting from the sides of the fuselage, enables fast deceleration, fast airspeed control and steep approach angles. The empennage is configured as a V-tail, while each wing half features four control surfaces of which the outermost one is used for flutter suppression (see Figure 3.4). A custom made actuator moves the surface with sufficient bandwidth. The two innermost control surfaces serve as high lift devices during takeoff and landing. The aircraft has two flutter modes. The first one gets unstable at $48.1 \frac{m}{s}$ and $7.95 Hz$, when the second bending and symmetric torsion modes
are coupled. The second flutter mode follows at $50.5 \frac{m}{s}$ and $6.42 HZ$ as an antisymmetric first bending form. Divergence occurs at $62.5 \frac{m}{s}$.

The aeroservoelastic dynamical model developed by the German Aerospace Center, DLR-Oberpfaffenhofen is based on a subsystem approach as outlined in the sequel (16; 1). The aerodynamics and the structural dynamics are developed separately and the interconnection forms the aeroservoelastic model (see Figure 3.5).

**Figure 3.5: Aeroelastic model**

### 3.2.3 Equations of Motion

First a finite element (FE) model is created to form the equations of motion for the aircraft (17). The FEM model goes under a modal decomposition and Guyan reduction (2), and a linear form is obtained:

\[
\begin{bmatrix}
\Phi_b^T & \Phi_f^T
\end{bmatrix}
\begin{bmatrix}
b_m \\
\eta_f
\end{bmatrix}
= P^{ext}(\omega)
\]

Equation 32 is explicitly split into a rigid body and flexible part (denoted by the subscripts $b$ and $f$) with modal matrices $\Phi_b$ and $\Phi_f$ respectively. Additionally $M$, $B$ and $K$ are the modal mass, damping and stiffness matrices respectively and $P^{ext}$ is the external excitation in modal coordinates. For the FLEXOP aircraft a 6 degree-of-freedom rigid body was used along with the flexible part consisting of 16 modes.

The linear model in Equation 32 is only valid for small perturbations, therefore the rigid body part is replaced by a non-linear one, describing the movement relative to a mean axes body reference frame. A common element in such applications is the Euler-Bernoulli-beam with added torsional effects. Based on some simplifying assumptions (1), the following form is obtained:

\[
\begin{bmatrix}
m_b \ddot{V}_b + \Omega_b \times V_b - T_{bE} g_b \\
j_b \Omega_b \times \Omega_b + j_b \Omega_b \\
M_{ff} \ddot{\eta}_f + B_{ff} \dot{\eta}_f + K_{ff} \eta_f
\end{bmatrix}
= \begin{bmatrix}
\Phi_b^T P^{ext}(\omega)
\\
\Phi_f^T P^{ext}(\omega)
\end{bmatrix}
\]

17
where $V_\text{b}$ and $\Omega_2$ are the velocity and angular velocity in the body frame. The mass distribution of the wing is assumed to be replaced by a concentrated mass system based on physical considerations. The 165 structural grid points, including the control surface deflections, are placed forward and after along the concentrated masses. The structural grid points have 6 degrees of freedom.

The external forces $\vec{F}_\text{ext}$ acting on the structure are dependent on the rigid body and flexible motion, as well as the atmospheric disturbances. The calculation of these unsteady aerodynamic forces is carried out by the Doublet Lattice Method (DLM) (9).

### 3.2.4 Aerodynamic forces

The unsteady aerodynamics is modeled with the subsonic Doublet Lattice Method (9), where the model is divided into aerodynamic panels. A short summary of the generalized aerodynamic model for the aerodynamic panels is given based on (18; 19).

The DLM results in the $AIC$ (Aerodynamic Influence Coefficient) matrices that relate the normal-wash vector $\vec{w}$ to the normalized pressure difference vector $\vec{p}$ about the panels as

$$\vec{p}_\text{panel} = [AIC_{\text{panel}}(\omega, V)] \vec{w},$$

where $\omega$ is the oscillating frequency and $V$ is the air speed. These two parameters are generally transformed into a single dimensionless parameter, the reduced frequency: $k = \frac{\omega}{2V}$, where $\bar{c}$ is the reference chord length. In order to relate the modal displacements to the normal-wash vector $\vec{w}$ and to transform the aerodynamic force to modal coordinates the so called generalized aerodynamic matrix (GAM) is defined as (see (18; 19) for more details)

$$Q\text{panel}(k) = \Phi^T T_{\Omega}^S [AIC_{\text{panel}}(k)] (D_1 + ikD_2) T_{\Omega} \Phi,$$

where $D_1$ and $D_2$ are the differentiation matrices, $S$ is the integration matrix and $T_{\Omega}$ is the interpolation matrix that projects the structural grid deformation on to the aerodynamic panels in form of their pitch and heave deformation (1). The GAM maps the modal deformation $\eta$ to the aerodynamic force distribution in modal coordinates. Note that the GAM matrices are obtained only over a discrete reduced frequency grid. However, time domain aeroelastic simulations require a continuous model. There are several methods to obtain such models (7), among these Roger’s rational function approximation (RFA) method (7) was applied for the underlying aircraft model. The resulting aerodynamic model is obtained in the form

$$Q\text{panel}(k) = Q_{\text{panel}_\text{q}} + Q_{\text{panel}_1} ik + Q_{\text{panel}_2} (ik)^2 + \sum_{i=1}^{n_p} \frac{ik}{b_i + ik},$$

where $Q_{\text{panel}_\text{q}}$, $Q_{\text{panel}_1}$ and $Q_{\text{panel}_2}$ stand for the quasi-steady, velocity and acceleration terms of the aerodynamic model. The $Q_{\text{panel}_1}$ terms take the lag behavior of the aerodynamic model into account. The poles of the lag states are given by $b_i$. $n_p$ number of poles are selected for each modal coordinate a priori. In order to appropriately capture the aerodynamic behavior $n_p = 30$ poles have been selected for the FLEXOP aircraft. This implies that the resulting aerodynamic model is of much higher
dimension than the structural model. At the same time, it is also clear that from an input-output point of view, the behavior can be well approximated with smaller dimensions, which will be later revealed by the model reduction algorithm. In a similar fashion, the GAM matrices for the control surface deflection $\delta_a$ can be also defined.

### 3.2.5 Model summary

The rigid body motion is modeled through the classical 6 degree-of-freedom description, which implies 12 state variables:

$$ x_{\text{rigid}} = [u, v, w, p, q, r, \phi, \theta, \psi, x, y, z]^T, $$

where $u, v, w$ are the body frame velocities, $p, q, r$ are the roll, pitch and yaw rates respectively, while $\phi, \theta, \psi$ are the roll, pitch and heading angles. The structural dynamics of the aircraft contains the first 16 structural modes and their time derivatives, which corresponds to a 32 state model. The aerodynamic model is constructed by selecting $n_p = 8$ poles for each structural coordinate. Therefore, the aerodynamic model consists of 480 lag states, and the final model has 522 state variables.

As illustrated in Figure 3.3, the aircraft has 12 control surfaces. Each control surface deflection together with its derivative and second derivative enters the model as control input, representing 36 input channels. Together with the additional thrust and air-brake signals the final model formulation has altogether 38 inputs.

The output signals of the aircraft can be divided into two groups. We assume that traditional sensors, i.e. accelerometers, rate gyros, pressure sensors and magnetometers provide rigid body related information. More specifically:

$$ y_{\text{rigid}} = [u, v, w, p, q, r, \phi, \theta, MACH, \beta]^T, $$

are available through measurements and can be used for synthesizing rigid body control laws. In addition to these measurements, dedicated flutter sensors will be also installed on the demonstrator aircraft. Accelerometers are placed at different cross sections of the wing on the forward and after structural grid points. Figure 3.6 shows possible configurations, from which only a few will be used. The optimal sensor selection for flutter suppression is currently in the focus of our research. In the present study we only used the $x$, $y$, directional acceleration and $x$, $y$, directional angular velocities of sensors $I_6$ and $R_6$, located at the tip of the wings. That is:

$$ y_{\text{flutter}} = [a_{xI6}^T, a_{yI6}^T, a_{xR6}^T, a_{yR6}^T]^T. $$
Consequently, the dimension of the measured output vector is $\mathbf{16}$.

Finally, the developed non-linear model has been trimmed and linearized at $N = 26$ different air speed values between $45$ and $70\text{ m/s}$, giving birth to a Linear Parameter Varying system model (20; 21):

$$\begin{align*}
\mathbf{\dot{x}}(t) &= A(\rho(t))\mathbf{x}(t) + B(\rho(t))\mathbf{u}(t) \\
\mathbf{y}(t) &= C(\rho(t))\mathbf{x}(t) + D(\rho(t))\mathbf{u}(t),
\end{align*}$$

with a grid-based representation:

$$\mathcal{G} = \{G_k | G_k = ss([A_k, B_k, C_k, D_k]), A_k = A(\rho_k), B_k = B(\rho_k), C_k = C(\rho_k), D_k = D(\rho_k)\}$$

Here, the scheduling parameter $\rho$ is the airspeed defined in the interval $\Gamma = [45 \text{ m/s}, 70 \text{ m/s}]$. $A_k \in \mathbb{R}^{524 \times 524}$, $B_k \in \mathbb{R}^{524 \times 33}$, $C_k \in \mathbb{R}^{16 \times 524}$ and $D_k \in \mathbb{R}^{16 \times 33}$ are the system matrices obtained from the linearization around trim point $\rho_k = 1, \ldots, N$.

The complexity and dimension of the developed description represent the main obstacle for exploiting the virtues of the described modeling chain and consequently using it for controller design (8). A model order reduction methodology is developed and applied to offer a remedy.

There are few restrictions for both the controller and model construction that are necessary for them to work smoothly together in the Matlab/Simulink configuration.

**General rules.** Both the model and the controller have to be implemented in the 2016b version of the MATLAB/Simulink program. All input/output signals are of standard double type, so no specific type conversions have to be performed.

Restrictions for the non-linear model:

**Model inputs $UM(Y_2)$**

- GearL
- GearR

---

D4.10 Release of the flight test results and models of the a/c for the community-y2019m11d30
WheelbrakeL
WheelbrakeR
AirbrakeL
AirbrakeR
Aileron1L
Aileron1R
Aileron2L
Aileron2R
Aileron3L
Aileron3R
Aileron4L
Aileron4R
Ruddervator1L
Ruddervator1R
Ruddervator2L
Ruddervator2R
Throttle

Model outputs $Y_M(U_d)$

- UTC Time
- Euler angles, in NED - $\phi, \theta, \psi$
- Acceleration in BF - $a_x, a_y, a_z$
- Position, in LLA - $X_{gr}, Y_{gr}, Z_{gr}$
- Angular Velocity in BF - $p, q, r$
- Course angle - $x$
- Velocity, in NED - $v_N, v_E, v_D$
- Absolute air velocity - $v_{IAS}$
- Angle of attack - $\alpha$
- Sideslip angle - $\beta$
- Altitude - $h$
- Static pressure - $p_{st}$
- Total pressure - $p_{total}$
- Indicated air speed - $v_{IAS}$

The following data is measured by each of the 12 IMUs installed in the wings according to Fig. 5:
- Acceleration - $z$
- Angle velocity - $\omega_x$
- Angle velocity - $\omega_y$
3.3 Coordinate definitions and reference values

For the sake of clarity the main reference values of the aircraft dimensions are listed in the following tables.

<table>
<thead>
<tr>
<th>Name/Description</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>aerodynamic center</td>
<td>0.76297</td>
<td>m</td>
</tr>
<tr>
<td>wing area half model</td>
<td>1.26839</td>
<td>m^2</td>
</tr>
<tr>
<td>mean aerodynamic chord (MAC)</td>
<td>0.37283</td>
<td>m</td>
</tr>
<tr>
<td>spanwise position of MAC</td>
<td>1.56563</td>
<td>m</td>
</tr>
<tr>
<td>aspect ratio full model</td>
<td>19.71</td>
<td>-</td>
</tr>
</tbody>
</table>
The aircraft uses several coordinate systems, attached to the wing, c.g. and to the undeformed aircraft global reference point.

**Global coordinate system**
- **abbr.**: GCS
- **location**: at the tip of the fuselage

**Wing coordinate system**

---

**Figure 3.9 Coordinate system reference points and directions**

The aircraft take-off mass is 65 kg.

The aircraft centre of gravity location is:
- x: 0.606 m
- y: -0.010 m
- z: -0.025 m

---

**Table: Aircraft take-off mass and centre of gravity location**

<table>
<thead>
<tr>
<th>Aircraft take-off mass</th>
<th>65 kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft centre of gravity location:</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>0.606 m</td>
</tr>
<tr>
<td>y</td>
<td>-0.010 m</td>
</tr>
<tr>
<td>z</td>
<td>-0.025 m</td>
</tr>
<tr>
<td>abbr.</td>
<td>WCS</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>location</td>
<td>aircraft midplane, at the wing leading edge</td>
</tr>
<tr>
<td>location wrt. GCS</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>0.8769 m</td>
</tr>
<tr>
<td>y</td>
<td>0 m</td>
</tr>
<tr>
<td>z</td>
<td>0 m</td>
</tr>
<tr>
<td>axis orientation wrt. GCS</td>
<td></td>
</tr>
<tr>
<td>xW</td>
<td>1,0,0</td>
</tr>
<tr>
<td>yW</td>
<td>0,1,0</td>
</tr>
<tr>
<td>zW</td>
<td>0,0,1</td>
</tr>
</tbody>
</table>

**Flight mech. CS**

<table>
<thead>
<tr>
<th>abbr.</th>
<th>FMCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>Origin at 25% MAC point</td>
</tr>
<tr>
<td>location wrt. GCS</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>0.76297 m</td>
</tr>
<tr>
<td>y</td>
<td>0</td>
</tr>
<tr>
<td>z</td>
<td>0</td>
</tr>
<tr>
<td>axis orientation wrt. GCS</td>
<td></td>
</tr>
<tr>
<td>xW</td>
<td>-1,0,0</td>
</tr>
<tr>
<td>yW</td>
<td>0,1,0</td>
</tr>
<tr>
<td>zW</td>
<td>0,0,-1</td>
</tr>
</tbody>
</table>

**0deg material axis (-0/-2 wing)**

<table>
<thead>
<tr>
<th>abbr.</th>
<th>A0</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>along the rear spar</td>
</tr>
<tr>
<td>axis orientation wrt. GCS</td>
<td></td>
</tr>
<tr>
<td>A0</td>
<td>0.2971, 0.9549, 0</td>
</tr>
</tbody>
</table>
4 Matlab/Simulink Implementation

4.1 Full order model

The FLEXOP_NonlinearModel_6_5_public_release folder contains the full order flexible aircraft model.

DISCLAIMER

This model was built by the Institute of System Dynamics and Control of the German Aerospace Center (DLR). The Institute of System Dynamics and Control does not guarantee, that the calculations of the model are correct. The model was set up purely by theoretical considerations and does not yet include any corrections with respect to flight or ground test data. Therefore, there might be a difference between the results of a simulation run and the flight test data.

The model was built in Matlab 2016b on Windows. To guarantee a correctly running model, it is recommended to run the model in Matlab 2016b.

The model developed for the FLEXOP project includes:

- equations of motion based on mean axes constraints, i.e. nonlinear rigid body motion and linear structural dynamics
- sensor systems: airdata probe, GPS
- control surface actuator models based on transfer functions
- fuselage mounted airbrakes
- engine model (state space system)
- gust inputs

Further information on the model can be found in the documentation of the FLEXOP project and the paper for the AIAA Aviation Conference 2018 in Atlanta "Aeroservoelastic Modeling and Analysis of a Highly Flexible Flutter Demonstrator" by Matthias Wuestenhagen et al. doi: 10.2514/6.2018-3150

Execution

The model can be initialized with the 'init.m' script. This file offers the opportunity to change simulation settings.

- Two different wing configurations can be chosen for 'name_data_file' (-0 wing -- stiff wing, -1 wing -- flutter wing).
- The initial indicated air speed is set by 'V_IAS'.
- The initial barometric altitude is set by 'h_baro'.
- The course angle is set by 'chi_K'.
- The vertical speed is set by 'der_h'.
- The roll attitude angle in case of a turning flight is set by 'Phi'.

After the execution of the 'init.m' script the aircraft simulation model (Simulink) opens automatically. The simulation can then be run by pressing the play button.

Uncertainties
The user is allowed to introduce parametric variations (uncertainties) within the model in the following parameters:

- indicated airspeed, relative factor \( f_{\text{ias}} = 1 \rightarrow \) no deviation
- aircraft height, absolute deviation \( h_{\text{dev}} = 0 \rightarrow \) no deviation
- \( m_{\text{delta}} \), additional mass
- \( lxx_{\text{delta}} \), additional \( lxx \)
- \( lyy_{\text{delta}} \), additional \( lyy \)
- \( lzz_{\text{delta}} \), additional \( lzz \)
- \( \text{cg}_{\text{shift}} \), change wrt reference co-ordinate system (\( x \): backwards, \( y \): to right wing, \( z \): upwards)
- change of modal damping/frequency
- \( \text{cs}_{\text{effect}} \), control surface effectiveness

The model also includes a placeholder block to implement both rigid body and flexible aircraft aeroelastic control laws.

The user has access to all sensor onboard the aircraft and can use all servos and actuators onboard. The flight control laws should respect the implementation constraints, so should be implemented in discrete time.

Figure 4.1 Main simulation view
The aircraft model is composed of 5 consecutive blocks, what follows the logical order in feedback control systems.

Input signals are grouped first and the corresponding trim values are applied on the control signals. Please note, that the aircraft model is valid only during airborne operation, the landing gears, and wheelbrakes are disconnected.
Figure 4.4 Input signal selection and trim

The raw input signals coming from pilot inputs or from the flight control system are passed through the actuator dynamics. Simple second order models with rate and deflection limits are used, with transport delay accounting for data transmission on the bus. Not just the deflections, but their first and second derivatives are also used in the simulation to properly simulate unsteady aerodynamics.

Figure 4.5 Actuator, engine and airbrake models
The core flight dynamics model includes unsteady aerodynamics, rigid body and flexible structural dynamics, with more than 300 states, including 30 flexible modes, and 288 aerodynamic lag states.

Figure 4.6 Aircraft rigid body and flexible dynamics model

The aircraft includes a number of sensors, what are all modeled with exact position and sensor dynamics where applicable, resulting in 95 outputs overall.
4.2 Reduced order model


The layout and usage of the model is the same, just the blocks inside the ‘Aeroelastic Aircraft Model’, as shown in Figure 4.6 Aircraft rigid body and flexible dynamics model) is different.

The reduced model is fully open, does not contain any black-box S-functions.

This model is placed in a separate folder FLEXOP_ReducedNonlinearModel. Running it is initiated by the init.m script placed in this folder.

DISCLAIMER

This model was built by the Institute for Computer Science and Control (SZTAKI), based on data provided by the Institute of System Dynamics and Control of the German Aerospace Center (DLR). The SZTAKI does not guarantee, that the calculations of the model are correct. The model was set up purely by theoretical considerations, involves special model order reduction assumptions and does not yet include any corrections with respect to flight or ground test data. Therefore, there might be a difference between the results of a simulation run and the flight test data.

The model was built in Matlab 2016b on Windows. To guarantee a correctly running model, it is recommended to run the model in Matlab 2016b.

The model developed for the FLEXOP project includes:
• equations of motion based on mean axes constraints, i.e. nonlinear rigid body motion and linear structural dynamics
• sensor systems: airdata probe, GPS
• control surface actuator models based on transfer functions
• fuselage mounted airbrakes
• engine model (state space system)
• gust inputs

Execution

The model can be initialized with the 'init.m' script. This file offers the opportunity to change simulation settings.

• One wing configurations available.
• The initial indicated air speed is set by 'V_IAS'.
• The initial barometric altitude is set by 'h_baro'.
• The course angle is set by 'chi_K'.
• The vertical speed is set by 'der_h'.
• The roll attitude angle in case of a turning flight is set by 'Phi'.

After the execution of the 'init.m' script the aircraft simulation model (Simulink) opens automatically. The simulation can then be run by pressing the play button.
5 Onboard Sensors

As described above the aircraft model contains several sensor measurements, including inertial, air data, GNSS sources for the rigid body motion. In addition 12 dedicated IMUs are placed inside the wings to measure the flexible behavior of the aircraft.

Figure 5.1 Overview of the Data Recording and Flight Control System

The overall system layout, with data recording and sensor – FCC communication is depicted in Figure 5.1. The corresponding datalogs inside the Flight Test Data logs also follow the same logic.
5.1 The flutter IMUs (flutterIMU)

Function
The main function of the flutter IMUs is, providing all the the desired inertial measurement data to the flutter control algorithm.

Every IMU contains a gyroscope, to measure the angular velocity in 3 perpendicular directions and 2 accelerometers, to measure the acceleration in a wide range with a fairly good resolution. One of the accelerometers measures the accelerations between +/- 16G in 3 perpendicular directions, the other has a +/- 70G acceleration range in the Z axis of the unit.

CAN bus is used for transferring data to the FCC at the maximum supported data rate by the protocol: 1Mbit/s. To minimize the delay caused by the data transfer, 2 separated CAN buses are used in each wing.

After manufacturing, the IMUs are not physically accessible, so every unit contains 2 exact same copies of the same circuit. This redundancy helps avoid the long and hard way of repair in case of unexpected failure.

The physical layout of the PCBs are shown in Figure 1 and Figure 2.

Inputs and outputs
Every unit has 1 interface to the outside world, a CAN bus at 1Mbit/s transfer rate. Via CAN bus the units can receive a data request message and respond with the measured values. This cycle runs in every 5 ms.

Technical and physical parameters
Physical parameters:
- Width: 30mm
- Length: 30mm
- Height: 10mm
- Weight: 8g

Technical parameters:
- Acceleration range: +/-16G in X,Y directions and +/-70G in Z direction
- Angular velocity: +/- 250°/s
- Voltage range: 5V-10V
- Current: approximately 50mA (depends on the state of operations)
Figure 5.2 The manufactured PCB of the first version of flutter IMU

Figure 5.3 The 3D preview of the flutter IMU with the actual modifications
D4.10 Release of the flight test results and models of the a/c for the community - y2019m11d30

Location of the IMUs inside the wing is described by the table below. Their measurements is implemented inside the mathematical model, by interpolating the surrounding mass points.

Table 5.1 Flutter IMU locations

<table>
<thead>
<tr>
<th>Inertial Measurement Unit (IMU)</th>
<th>IMU dimension</th>
<th>Wing coordinate system:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>28mm x 28mm</td>
<td>*see Definitions sheet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>label</th>
<th>x [mm]</th>
<th>y [mm]</th>
<th>z [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>415</td>
<td>1061</td>
<td>/</td>
</tr>
<tr>
<td>R2</td>
<td>602</td>
<td>1061</td>
<td>/</td>
</tr>
<tr>
<td>R3</td>
<td>789</td>
<td>2121</td>
<td>/</td>
</tr>
<tr>
<td>R4</td>
<td>933</td>
<td>2121</td>
<td>/</td>
</tr>
<tr>
<td>R5</td>
<td>1164</td>
<td>3183</td>
<td>/</td>
</tr>
<tr>
<td>R6</td>
<td>1260</td>
<td>3182</td>
<td>/</td>
</tr>
<tr>
<td>L1</td>
<td>415</td>
<td>-1061</td>
<td>/</td>
</tr>
<tr>
<td>L2</td>
<td>602</td>
<td>-1061</td>
<td>/</td>
</tr>
<tr>
<td>L3</td>
<td>789</td>
<td>-2123</td>
<td>/</td>
</tr>
<tr>
<td>L4</td>
<td>933</td>
<td>-2120</td>
<td>/</td>
</tr>
<tr>
<td>L5</td>
<td>1164</td>
<td>-3184</td>
<td>/</td>
</tr>
<tr>
<td>L6</td>
<td>1260</td>
<td>-3183</td>
<td>/</td>
</tr>
</tbody>
</table>

5.2 The servo health monitoring unit (SHM)

Function

The main function of the servo health monitoring units is to monitor the state of servos and send the servo deflection and temperature values to the (Flight Control Computer) FCC. Two kinds of data are supplied: the units measure the temperature of the servos and take samples from the servo angle feedback potentiometers. Both sensors provide analog signals, so the unit contains impedance matching components. The analog signals are attached to a microcontroller. The microcontroller takes samples from the analog signals and sends the values via CAN bus after a received request message.
Similar to the flutter sensing IMUs, the data transfer rate reaches 1 Mbit/s, which guarantees the fast data exchange between the hardware elements.

The physical layout of the PCB is shown in Figure 5.5 and Figure 5.6.

**Inputs and outputs**

The temperature sensors are built in the unit, so there is not extra connector and wire required between the sensor and the PCB. The analog signal from the potentiometers attached to the unit directly.

The units have a CAN communication interface, at 1 Mbit/s.

**Technical and physical parameters**

Physical parameters:
- Width: 20mm
- Length: 40mm
- Height: 5mm (exclude the thermal sensor)
- Weight: 5g

Technical parameters:
- Temperature range: -40°C to +125°C
- Potentiometer angle resolution: 12 bits
- Voltage range: 5V-10V
- Current: approximately 50mA (depends on the state of operations)

![Figure 5.5 The upper side of the servo health monitoring PCB](image-url)
5.3 xSens

Function
The xSens MTi 710 sensor measures inertial and position data (Figure 5.7). These data are used by the rigid body controller. To measure these data the sensor includes a GPS, a magnetometer, a gyroscope and an accelerometer. From these measurements the sensor is able to estimate other type of data (for example Euler angles or quaternion) and the user is able to choose one or more out of them.

Inputs and outputs
Currently the following data are streamed: UTC time, Euler angles, acceleration, angular velocity, position and velocity at the frequency of 400 Hz to reduce the time delay. The controller decimalize the measurement data to 200 Hz of frequency.

The outputs are sent via RS232. Every data packet has a header and a checksum. To minimize the time delay the baud rate is set to the available maximum that is 921600 bit per second (bps).

After the sensor boots up, it is recommended to configure the xSens to be sure that it sends the required outputs. To configure the sensor it is needed to send configuration messages. After the configuration the sensor starts streaming the measurement data.

Technical and physical parameters
Physical parameters:
- Size: 57×42×24 mm
- Mass: 58 g

Technical parameters:
- Measurement range:
  - Acceleration: ±50 m/s²
  - Angular velocity: ±450 °/s
  - Velocity: 500 m/s
- Voltage range: 4.5-34 V
- Current:
  - 106 mA at 5 V
  - up to 200 mA at startup

### Table 5.2 Rigid body IMU location

<table>
<thead>
<tr>
<th>Positions in Reference to the Wing Coordinate system</th>
<th>IMU Xsens Fuselage [CG]</th>
</tr>
</thead>
<tbody>
<tr>
<td>X [m]</td>
<td>Y [m]</td>
</tr>
<tr>
<td>-0.087</td>
<td>0.000</td>
</tr>
</tbody>
</table>

### 5.4 Micro Air Data System

**Function**

This sensor is used to measure the Angle of Attack and the Angle of Sideslip which are used by the rigid body controller. The Micro Air Data System consists of two parts: a multi hole Air Data Probe and a micro Air Data Computer (see Hiba! A hivatkozási forrás nem található.). The probe has 5 holes and a static ring. This probe is connected to the micro Air Data Computer through pipes. The Computer measures the pressure differences and calculates the sensor’s output.
**Inputs and outputs**

The sensor sends the Angle of Attack, the Angle of Sideslip, the velocity, the altitude, the static pressure, the total pressure, the date and the time. The last two are inaccurate because GPS is not connected to the device. These data are sent at the frequency of 100 Hz, which is the available maximum.

The sensor sends the whole measurement data as a string via RS232. The speed of the transmission is 460800 bit per second.

After the sensor boots up it is recommended to configure the sensor to be sure that its settings are the required ones.

After the successful modification of the settings the sensor reboots and starts streaming data.

**Technical and physical parameters**

**Physical parameters:**
- Size: 66×79×33 mm
- Mass: 135 g

**Technical parameters:**
- Measurement range:
  - Velocity: 359 m/s
  - Angle of Attack: ±20°
  - Angle of Sideslip: ±20°
  - Altitude: -298 – 18293 m
- Voltage range: 12-15 V
- Current: maximum 270 mA at 12 V

This is the main part of the deliverable and should explain clearly how the results were achieved, including diagrams or pictures to illustrate technical/scientific points.

In case the actual deliverable is not a document, but a prototype or test results etc., it needs to be documented with a user manual, photos and description of location, specifications, test scenario or any other appropriate information.

<table>
<thead>
<tr>
<th>Table 5.3 Airdata probe location</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Positions in Reference to the Wing Coordinate system</strong></td>
</tr>
<tr>
<td>Air Data Probe [CG]</td>
</tr>
<tr>
<td>-0.899</td>
</tr>
<tr>
<td>0.000</td>
</tr>
<tr>
<td>-0.100</td>
</tr>
<tr>
<td>Air Data Probe [Tip]</td>
</tr>
<tr>
<td>-1.436</td>
</tr>
<tr>
<td>0.000</td>
</tr>
<tr>
<td>-0.05</td>
</tr>
</tbody>
</table>

**5.5 Shape Monitoring System**

This chapter describes the subsystems within the fiber optical monitoring system, including their definitions, business goals, business objectives, context and/or capabilities.

**Introduction to Fiber Bragg Gratings**

A fiber Bragg grating (FBG) is a type of distributed Bragg reflector constructed in a short segment of optical fiber that reflects particular wavelengths of light and transmits all others. This is achieved by creating a periodic variation in the refractive index of the fiber core, which generates a wavelength...
A fiber Bragg grating can therefore be used as an inline optical filter to block certain wavelengths, or as a wavelength-specific reflector.

![Image of a fiber Bragg grating](image)

**Figure 5.9 Schematic representation of the operational principle of FBGs**

The fundamental principle behind the operation of a FBG, is Fresnel reflection. Where light traveling between media of different refractive indices may both reflect and refract at the interface.

The grating will typically have a sinusoidal refractive index variation over a defined length. The reflected wavelength ($\lambda_B$), called the Bragg wavelength, is defined by the relationship:

$$\lambda_B = 2n_e \Lambda$$

*Eq. 5-1*

where $n_e$ is the effective refractive index of the grating in the fiber core and $\Lambda$ is the grating period. The effective refractive index quantifies the velocity of propagating light as compared to its velocity in vacuum. $n_e$ depends not only on the wavelength but also (for multimode waveguides) on the mode in which the light propagates. For this reason, it is also called modal index.

![Image of power distribution](image)

**Figure 5.10 Power distribution of reflected spectrum from a FBG with wavelength $\lambda_B$**
The wavelength spacing between the first minima (Hiba! A hivatkozási forrás nem található.), or the bandwidth ($\Delta \lambda$), is (in the strong grating limit) given by:

$$\Delta \lambda = \left[ \frac{2\delta n_0}{\pi} \right] \lambda_B$$  \hspace{1cm} Eq. 5.2

where $\delta n_0$ is the variation in the refractive index ($\eta_3 - \eta_2$), and $\eta$ is the fraction of power in the core.

In practice this can be applied as follows; using a broadband light source sending a signal spectrum into the fiber, only certain wavelengths according to the FBG sensors are reflected. By means of spectrometer these wavelengths – and their shifts due to strain or temperature change – can be monitored and related to the strain levels picked up by the FBG sensors.

Figure 5.11 Schematic representation of the operation of several FBGs within a single optical fibre

5.5.1 Monitoring system operation

Along with the required number of sensors, one of the first questions that should be clarified before the definition of the system layout and characteristics is the way of operation. In that view limitations like physical, power and temperature should be clarified. Additionally issues like connectivity and synchronization with other devices and data handling and usage should also be addressed.

The objective of FLEXOP project is to build a UAV with several sets of wings to test with several flight tests. The scope of the FBG system is to monitor the developed strains and forces on the wings during each flight tests. Having this in mind it is easily understood that the monitoring system should:

i. Be able to monitor all FBG sensors on the wings
ii. Store all generated data
iii. Fit within the available space in the fuselage of the UAV
iv. Have a power consumption that is as low as possible
v. Be able to operate within the temperature range of the flight test
vi. Be able to receive a synchronization signal
For the needs of these objectives INASCO is in close collaboration with all FLEXOP partners and due to the fact that the monitoring system is being developed in a parallel way with the UAV this process is ongoing and iterative. However, issues that have to do with the size, power and temperature limitations have already been addressed. Also a first estimation of the required sensors has allowed for the selection of the most suitable interrogator type and number for the needs of this task.

Due to the required high number of interrogated sensors it is anticipated that the respective amount of generated data will be proportionally high. For this reason it is decided that the monitoring system will have to store all the generated data from each flight at a storage device instead of transmitting them lively to the ground base. The storage device will either be exchanged at the end of each flight test or have the appropriate size to store the data from all test flights. This approach (storing the data) allows also for a much simpler connection of the interrogation units within the UAV and less power consumption due to the limited amount of date transmitted from the UAV to the ground base.

Based on the above a general overview of the system layout has been identified and it is described in more details in the following paragraphs.

5.5.2 General overview of the sensing system layout

The diagram below sketches the general system layout for the fiber optic monitoring system.

![Schematic representation of the FBG based monitoring system layout](image)

As it is well understood from Hiba! A hivatkozási forrás nem található. all the components that will be used for the monitoring of the strains and forces on the wings of the UAV are divided into two major categories:
i. Fuselage components and
ii. Wing components

The wing components contain all the optical fibers that have the FBG sensors and the required connectors that will be attached / integrated to the wings while the fuselage components contain all the remaining equipment necessary to interrogate the sensors and store the data. It should be mentioned here that the components included in the flight control computer are not considered part of the monitoring system but are necessary in order to generate the synchronization signal that will be used for the post processing of the data.

Each wing will have its own set of sensors that will be integrated to the wing during its manufacturing process. The rest of the monitoring system (interrogation unit, PCs, HDDs) will be installed and secured in the fuselage at specified locations after the manufacturing of the fuselage. Each wing will come already equipped with sensors and the optical fibers that will go out of the root rib will be routed and connected to the interrogator(s) after the assembly of each pair of wings. Due to the high amount of sensors that is currently expected for the needs of this monitoring task space, weight and power for up to **four (4) interrogation systems** has been reserved that is able to monitor up to **500 sensors**. However, the exact number of required systems will be defined with the progress of the design works and the definition of the exact number of required sensors.

Table 5.4 FBG locations and types inside the wings (Active FBGs during FT3 – left hand wing)

<table>
<thead>
<tr>
<th>Gator</th>
<th>Channel</th>
<th>Wing Skin</th>
<th>Fiber ID</th>
<th>FBG No.</th>
<th>Type of FBG</th>
<th>Spanwise Position [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
<td>Lower</td>
<td>OB,L</td>
<td>1</td>
<td>0° – Ros. 9</td>
<td>1.285</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>0° – Ros. 10</td>
<td>1.493</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0° – Ros. 11</td>
<td>1.641</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>0° – Ros. 13</td>
<td>1.948</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>0° – Ros. 14</td>
<td>2.105</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>0° – Ros. 15</td>
<td>2.336</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>Lower</td>
<td>B20-22,L</td>
<td>1</td>
<td>90° – Ros. 17</td>
<td>2.694</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>0° – Ros. 17</td>
<td>2.694</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>45° – Ros. 17</td>
<td>2.694</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>90° – Ros. 18</td>
<td>2.951</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>0° – Ros. 18</td>
<td>2.951</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>45° – Ros. 18</td>
<td>2.951</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>Temp. 9</td>
<td>2.951</td>
</tr>
<tr>
<td></td>
<td>G1</td>
<td>Lower</td>
<td>B23-24,L</td>
<td>1</td>
<td>90° – Ros. 19</td>
<td>3.153</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>0° – Ros. 19</td>
<td>3.153</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>45° – Ros. 19</td>
<td>3.153</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>90° – Ros. 20</td>
<td>3.365</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>0° – Ros. 20</td>
<td>3.365</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>45° – Ros. 20</td>
<td>3.365</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>Temp. 10</td>
<td>3.365</td>
</tr>
<tr>
<td></td>
<td>C4</td>
<td>Upper</td>
<td>B23-24,U</td>
<td>1</td>
<td>90° – Ros. 19</td>
<td>3.153</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>0° – Ros. 19</td>
<td>3.153</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>45° – Ros. 19</td>
<td>3.153</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>90° – Ros. 20</td>
<td>3.365</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>0° – Ros. 20</td>
<td>3.365</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>45° – Ros. 20</td>
<td>3.365</td>
</tr>
</tbody>
</table>
### Table 5.5 Location and type of FBGs (Active FBGs during FT3 – right hand wing)

<table>
<thead>
<tr>
<th>Gator</th>
<th>Channel</th>
<th>Wing Skin</th>
<th>Fiber ID</th>
<th>FBG No.</th>
<th>Type of FBG</th>
<th>Spanwise Position [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2</td>
<td>C1</td>
<td>Lower</td>
<td>45A,L</td>
<td>1</td>
<td>45° – Ros. 1</td>
<td>0.227</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>45° – Ros. 2</td>
<td>0.360</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>Lower</td>
<td>0A,L</td>
<td>1</td>
<td>0° – Ros. 1</td>
<td>0.227</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>0° – Ros. 2</td>
<td>0.360</td>
</tr>
<tr>
<td></td>
<td>C3</td>
<td>Lower</td>
<td>T,L</td>
<td>1</td>
<td>Temp. 1</td>
<td>0.388</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>Temp. 2</td>
<td>0.593</td>
</tr>
<tr>
<td></td>
<td>C4</td>
<td>Lower</td>
<td>90A,U</td>
<td>1</td>
<td>90° – Ros. 1</td>
<td>0.227</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>90° – Ros. 2</td>
<td>0.360</td>
</tr>
<tr>
<td></td>
<td>C5</td>
<td>Upper</td>
<td>0A,U</td>
<td>1</td>
<td>0° – Ros. 1</td>
<td>0.227</td>
</tr>
</tbody>
</table>

D4.10 Release of the flight test results and models of the a/c for the community-y2019m11d30
| C1 | Lower | 90B,L | 2 | 0° – Ros. 2 | 0.360 |
|    |       |       | 4 | 0° – Ros. 4 | 0.586 |
|    |       |       | 5 | 0° – Ros. 5 | 0.688 |
|    |       |       | 6 | 0° – Ros. 6 | 0.875 |
|    |       |       | 7 | 0° – Ros. 7 | 1.005 |
| C2 | Lower | B23-24,L | 1 | 90° – Ros. 9 | 1.285 |
|    |       |       | 2 | 90° – Ros. 10 | 1.493 |
|    |       |       | 1 | 90° – Ros. 19 | 3.153 |
|    |       |       | 2 | 0° – Ros. 19 | 3.153 |
|    |       |       | 3 | 45° – Ros. 19 | 3.153 |
|    |       |       | 5 | 0° – Ros. 20 | 3.365 |
|    |       |       | 6 | 45° – Ros. 20 | 3.365 |
| C3 | Upper | 45B,U | 1 | 45° – Ros. 9 | 1.285 |
|    |       |       | 2 | 45° – Ros. 10 | 1.493 |
|    |       |       | 3 | 45° – Ros. 11 | 1.641 |
| C4 | Upper | 90B,U | 1 | 90° – Ros. 9 | 1.285 |
|    |       |       | 2 | 90° – Ros. 10 | 1.493 |
|    |       |       | 3 | 90° – Ros. 11 | 1.641 |
| C5 | Upper | B20-22,U | 1 | 90° – Ros. 17 | 2.694 |
|    |       |       | 2 | 0° – Ros. 17 | 2.694 |
|    |       |       | 3 | 45° – Ros. 17 | 2.694 |
|    |       |       | 4 | 90° – Ros. 18 | 2.951 |
|    |       |       | 5 | 0° – Ros. 18 | 2.951 |
|    |       |       | 6 | 45° – Ros. 18 | 2.951 |
|    |       |       | 7 | Temp. 9 | 2.951 |
Figure 5.13 Example of raw strain readings during the flight

Sensor locations and types are important to associate the corresponding raw strain measurements with physical behavior of the wing when analyzing the flight test data.
6 Description of the Flight Test

Flight test data of the demonstrator is also included with the release. There are two complete flights included, denoted as FT03 and FT04. They correspond to flight on 16.10.2019 and 23.10.2019. The first one includes autopilot initiated rigid body identification signals, the second one mostly includes pushover-pull-ups and constant bank angle maneuvers, flown manually by the pilot to assess the wing behavior in high load factor maneuvers – to demonstrate wing root bending moment reduction on -2 wing with respect to the reference values recorded in this flight.

The data packages are denoted as FLEXOP_FT03_FTD_y2019m10d16.zip and FLEXOP_FT04_FTD_y2019m10d23.zip. Both of them includes the Flight test card, the raw FCC log in Matlab .mat format (like FLEXOP_FT03_FCC_y2019m10d16.mat), and the raw FBG readings in .csv format (like FLEXOP_FT03_Gator1_y2019m10d16.csv). The corresponding data post processing tools are included in the LogPlotter folder.

The specific details of Flight test 3 are the following:

<table>
<thead>
<tr>
<th>Flight Test 1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
</tr>
<tr>
<td>Engine Start/Stop Time</td>
</tr>
<tr>
<td>Test</td>
</tr>
<tr>
<td>Limitations</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Configuration</td>
</tr>
<tr>
<td>Success?</td>
</tr>
<tr>
<td>Test Point List:</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The goal of the Test Flight 1.3 was to do rigid mode identification manoeuvres, airspeed calibration checks and aeroelasticity measurements with the Fibre-Bragg system. First time the flight was attempted on 11.09.2019, but the flight was cancelled due to two unsuccessful takeoff attempts, during which the fuselage was damaged by a light on the side of the runway. The flight was rescheduled to 16.10.2019., after the repair was done.
Data gathered during the identification manoeuvres was compared to simulation results by DLR-SR. In the simulation inputs to control surfaces were fed into the simulation model. The model was trimmed for steady horizontal flight: trim altitude was first entry of altitude during the manoeuvre and trim speed was the first entry of speed during the manoeuvre. Some of the raw flight test data comparison is displayed in Figure 7.2 - Figure 7.9. It was noted that the pitch rate is very well recreated by the simulation, whereas altitude change during phugoid manoeuvre or sideslip during dutch-roll was not so accurate.

The specific details of Flight test 4 are the following:

<table>
<thead>
<tr>
<th>Date</th>
<th>23.10.2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine Time</td>
<td>Start/Stop 15:22 – 15:41 (duration 19min)</td>
</tr>
<tr>
<td>Test</td>
<td>Aeroelastic Tests</td>
</tr>
<tr>
<td>Test Objective</td>
<td>Aeroelastic tests (circles and pushover-pull-ups).</td>
</tr>
</tbody>
</table>
| Limitations | V_min = 25m/s, V_max = 53m/s  
               | H_min = 150m, H_max = 300m   |
| Configuration | TOW = 65kg, CG = 606mm. |
| Success?    | Yes                  |
Flight Test 1.4 was almost solely dedicated to gathering data about wing elasticity under flight loads. This was done with two different manoeuvres: the pushover-pull-up manoeuvres and circles with increasing bank angle.

The protocol of the pushover-pull-up manoeuvre was as follows:

1. Trim at 40m/s.
2. Pull-up slightly to drop the speed to 35m/s.
3. Pushover until flight path angle was around -30deg.
4. When 45m/s is reached, pull-up with constant elevator angle.
5. When aircraft decelerates to 30m/s, level out.

This had to be done with pilot not knowing the elevator deflection nor the flight path angle, because the manoeuvre was too fast for the information to be communicated from the ground control station. However, the resultant data was of good quality and seven of such test points with average load factors reaching around 2.5 were achieved.

Circles with bank angles of 40-60deg followed, with couple of small sideslip manoeuvres at the end.

The pushover-pull-up manoeuvres were again used as input for the simulation environment by DLR-SR. Comparison in between the actual flight test data and simulation results displayed a good correlation in indicated airspeed and pitch rate (Figure 7.11 and Figure 7.1).
Figure 6.2 Airspeed and altitude graphs (FT04)

Figure 6.3 Load factor graph (FT04)
7 Flight Test Data

Flight test results also confirmed the high fidelity and accuracy of the model based predictions. Special, system identification related maneuvers were performed during the flight test campaign. A comparison was done to see the longitudinal-, lateral-dynamics and also the control effectiveness.

7.1 Log-plotter: Overview

This project offers two ways to process and plot the parsed logs from the FLEXOP aircraft. For both methods you need MATLAB R2016b or later.

1.) The script method
This method utilizes the post_processor.m and plotter.m scripts.

1.1.) Post_processor
This is a log post processing tool to make the data plotting procedure easier. Basically, it creates separate .mat files for the main types of data saved on the FCC. This makes it faster to load the data you specifically need. Moreover, there are some calculations carried out on the PWM, PPM, SHM and IMU data.

1.1.1.) Features
1. Imports a parsed .mat file (e.g. 24.mat)
   • This can be edited in the "Load the data" section
2. Calls the servoparser function to convert the SHM values to degrees
3. Deletes the unnecessary variables
   • This is also modifiable in the "Delete the obsolete variables" part
4. Creates three time vectors (please note the minor inconsistency and missing data in the logfiles):
   • timer_1: generated time vector with constant spacing
   • timer_2: time vector from xSens time
   • timer_3: sample instances
5. Trims the logs to the supposed flight time (if it doesn't detect a takeoff, no trimming is done)
6. Filters the RX_MUX data
7. Converts the IMU measurements to SI values and renames them according to "wing_IMUs.jpg"
8. Renames the PWM, SHM, and PPM variables to meaningful names, based on the Wing_ref.xlsx table
9. Saves the fields to separate .mat files (like 24_IMU.mat)

1.1.2.) How to use
1. Write the number of the log you want to process into “filename” variable
2. Run the script
1.2.) Plotter
This script is useful to plot the processed logs. Note, that you have to run the post_processor script before, to get the necessary files for this script.

1.2.1.) How to use
1. Write the number of the log you want to plot according to the "Setup and how to use section"
2. Set the flags to turn on (1) or off (0) the variables to be plotted
3. Select the time vector you want to plot against
4. Run the script

You can run the sections of the script by positioning the cursor inside and pressing ctrl+enter (Note, that the flag corresponding has to be set)

1.3) Xsens_high_pass
This script high-pass filters the xSens normal acceleration data, then checks the correlation between the filtered xSens and wing mounted IMU values. Moreover, it rotates the gyros to the body coordinate system, since the wing has a 20 deg sweepback relative to the body y axis. Then checks the correlation here as well. This is necessary since the wing mounted IMUs have a high-pass filter implemented in them to remove gravity vector related offsets. The high-pass filter has the form:
num=[0.99968593938935957 -0.99968593938935957]; %high pass filter numerator
den=[1 -0.99937187877871914]; %high pass filter denominator
iir_filt=tf(num,den,0.001); %filter is implemented with 0.001 s sample time

The following plots are created:

<table>
<thead>
<tr>
<th>Plot #</th>
<th>plotted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vertical acceleration</td>
</tr>
<tr>
<td>2</td>
<td>Vertical acceleration difference between filtered Xsens and IMU01</td>
</tr>
<tr>
<td>3</td>
<td>Normal acceleration (with IMU-s 01-04)</td>
</tr>
<tr>
<td>4</td>
<td>Normal acceleration (with IMU-s 05-08)</td>
</tr>
<tr>
<td>5</td>
<td>Normal acceleration (with IMU-s 09-12)</td>
</tr>
<tr>
<td>10</td>
<td>Single-Sided Amplitude Spectrum of vertical acceleration(t) (IMU1, 2, 3)</td>
</tr>
<tr>
<td>20</td>
<td>Angular rate</td>
</tr>
<tr>
<td>21</td>
<td>Angular rate in body coordinate system</td>
</tr>
<tr>
<td>22</td>
<td>Y Angular rate in body coordinate system</td>
</tr>
<tr>
<td>23</td>
<td>X Angular rate in body coordinate system</td>
</tr>
<tr>
<td>24</td>
<td>Single-Sided Amplitude Spectrum of angular rate(t)</td>
</tr>
</tbody>
</table>

1.3.1) How to use
Modify the log number in the "logfile" variable at the beginning of the script to the one you want to examine, then run the script.
2.) The GUI method
You have to launch the Plotter_app, which is a basic MATLAB GUI app for plotting.

2.1.) Features
- It processes the "raw" (but parsed) .mat log file
- You can select the fields to be loaded
- You can plot up to 4 signals in the same figure
- Opening the figure externally is also possible
- 3D trajectory can be opened in an external figure
- Run the xsens_high_pass script on the loaded data and examine the plots
- View the “wing_IMUs.jpg” to help with the positions of servos and IMUs

2.2.) How to use
1. Select the fields you want to plot later on with the checkboxes
2. Click on "Select log file" button and select the file (only the xx.mat)
3. Wait until the dropdown lists are filled with the signals, then select the ones you want to plot in the ‘Signals to be plotted’ dropdown lists left to the figures
4. Select the time vector to plot against
5. Hit the "Plot!" button to draw them
6. if you want, click open them in an external figure with the button below the dropdown lists
7. Click the “3D trajectory” button to view the trajectory in an external figure
8. Click on the “xSens HPF” button to run the xsens_high_pass script
9. Hit the “Help” button to view the “wing_IMUs” image

7.2 Data Analysis
In the comparison of ‘Simulation vs. Flight Test’ maneuvers were extracted from flight test data and the corresponding inputs to control surfaces fed in simulation model. The simulation model was trimmed for steady horizontal flight, with the following assumptions: trim altitude is the first entry of altitude during the maneuver, as well as trim speed is also the first entry of speed.
Figure 7.1 elevator pulse input to assess phugoid mode (time vs. pitch rate)

Figure 7.2 time vs. altitude for phugoid excitation with elevator
Maneuver: elevator_pulse, Variable: Y_alpha

Figure 7.3 angle of attack response for phugoid excitation with elevator pulse

Maneuver: elevator_doublet2, Variable: Y_q

Figure 7.4 pitch rate response to elevator doublet
D4.10 Release of the flight test results and models of the a/c for the community-y2019m11d30

Figure 7.5 Dutch-roll excitation with rudder doublet, sideslip angle response

Figure 7.6 Dutch-roll excitation with rudder doublet, yaw rate response
Figure 7.7 dutch-roll excitation with rudder doublet, roll rate response

Figure 7.8 sideslip command by pilot input, sideslip angle response
Maneuver: sideslip\_1, Variable: Y\_r

Figure 7.9 sideslip command by pilot input, yaw rate response

Maneuver: jurij\_man1, Variable: Y\_alt

Figure 7.10 Pushover-pull-up maneuver, for load alleviation testing, altitude response
We have identified number of issues what have been partially corrected and addressed during data analysis and evaluation related to the flight test data:

- It is difficult to work with uncalibrated control surfaces. Even though the lookup tables have been captured to map the PWM input to commanded deflection angles, and also the measured deflections by the servo health monitoring units the mapping is not perfect, a drift can be seen on the signals and an offset in certain cases (what might be caused by replacing a servo).

- The IMUs are slightly differently oriented in the actual wings then in the idealistic model. Not just the flexible deformation of the wing corrupts the assumptions but also the physical imprecision to align the sensors with the front and rear spars.

- Due to flight test data logging issues onboard the FCC, in a few instances sudden jumps can be seen in the signals, where data is missing from the logs. This has no physical background, rather it is an artifact how we resample the signals to get rid of the missing measurements. This must be improved in the future.

- Even though the simulation model predicts the aerodynamic forces, there is large uncertainty in the control surface effectiveness, since it is very sensitive to geometry and initial assumptions.

- During simulations the assumption is we initialize the aircraft from trim, while in reality, especially with manual flight the aircraft is in transient mode leading to a difference in initial state between the model vs. flight test data.
We can only assume steady wind, continuous (Dryden) and discrete (1-cos) gust disturbances in simulation, while in reality the external disturbances are more random and we do not have the necessary instruments onboard the aircraft to perfectly reconstruct the external wind field. The FBG strain data is also included with the data package. Raw strain measurements can be seen in Figure 7.12, where the drift, most likely caused by thermal effects, on the data is also noticeable.

![Raw Strains - Exp. 301, W-2, LH - 45A,L](image)

**Figure 7.12 Raw FBG measurements**

The location of the sensors is described in Table 5.4 and Table 5.5, while the sensor geometry is depicted in Figure 7.13. It must be noted, that significant post processing is required to obtain calibrated strain data, what is not included with this distribution. A few important aspect of developing such post processing pipeline:

- **Normalisation**
  - Use of airspeed and mass at 1G, to identify lift distribution
  - Apply lift and gravity forces to get numerical 1G strains

- **Pull-Up Strain**
  - The peak and plateau strains are averaged and subtracted
  - Their difference is added to the numerical 1G strains
  - The corresponding load factor and airspeed are also found

To be able to obtain correct results one might utilize the linear static FEM model. This would allow building a virtual testing platform capable of accurately simulating loads, shapes, strains. Based on this, using the data from static test and GVT, calibration can be done. Within FLEXOP theoretical and
Algorithmic tools for static, dynamic and flight strain processing have been performed. This allows also reliable calculation of deflection, not just strains. It must be also noticed, that the sensor layout and calibration dataset does not seem suitable for twist calculation on stiff and large aspect ratio wings.

Figure 7.13 FBG sensor locations

Figure 7.14 Pull-up maneuver with corresponding load factor and velocity
Figure 7.15 Measured strain during maneuver
8 Deviations and Conclusion

The consortium, in the Description of Work committed to release flight test data, more specifically according to D5.3 the description of Mandatory publicly available datasets are the following:

"According to plan the following set of data and simulation files will be available publicly and fall under the procedure described above:

According to the GA [3] Part B page 27:

To comply with consortium confidentiality aspects, especially related to aeroelastically tailored wings, only the following set of data and simulation files will be available publicly:

A high fidelity simulation model of a flexible aircraft (-1) compiled partially as a black box model within Matlab/Simulink environment,

A reduced order control design oriented flexible aircraft model (fully open) within Matlab/Simulink environment of the same (-1) configuration,

A complete set of flight test data (several flights) of the corresponding (-1) wing configuration with all sensory data and actuator commands."

Since the project have not achieved flight with the -1 wing, it is not possible to share flight test data about it. On the other hand, to be able to compare simulation models and flight test data, the corresponding -0 datasets and high-fidelity models are included. Also, to be able to assess the control design and flutter mitigation measures, the high-fidelity and the control oriented reduced-order models of the -1 wing are included in the public dataset. Two flight test datasets (FT03 and FT04) are released, with a very rich set of sensors, including wing IMUs and FBGs, with the corresponding tools to assess them by external partners.

Project partners combined their tools and methods in a very efficient and impressive way during the project to capture and assess the datasets now shared with the public. This must be very valuable for the whole aeroelastic community.