AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-1970

Active Flutter Mitigation Testing on the FLEXOP Demonstrator Aircraft

Béla Takarics and Bálint Patartics and Tamás Luspay and Péter Bauer and Bálint Vanek
SZTAKI. Budapest, Kende u. 13-17, 1111, Hungary

Christian Rößler and Julius Bartasevicius and Sebastian Köberle and Daniel Teubl and Mirko Hornung
Technical University of Munich, Institute of Aircraft Design, Boltzmannstraße 15, Garching, München, 85748, Germany

Sergio Waitman and Andres Marcos
University of Bristol, Bristol, BS8 1TR, UK

Manuel Pusch and Matthias Wüstenhagen and Thiemo Kier and Gertjan Looye
German Aerospace Center DLR -Institute of System Dynamics and Conrol, Wessling, 82234, Germany

Yasser M. Meddaikar
German Aerospace Center DLR - Institute of Aeroelasticity, Göttingen, 37073, Germany

The paper details the research and corresponding implementation and testing steps of the FLEXOP demonstrator aircraft. Within the EU funded project an unmanned demonstrator aircraft is built to validate the mathematical modelling, flight control design and implementation side of active flutter mitigation. In order to validate the different methods and tools developed in this project, a flight test campaign is planned, in which the design and manufacturing of stiff wings (-0), are compared with very flexible wings (-1) with active flutter control, to see the overall benefit vs. risk of such technology. The mathematical models of the aircraft are first developed using FEM and CFD tools, what are later reduced by model order reduction techniques. The high-fidelity models are updated using Ground Vibration Test results. Manufacturing tolerances and variations in aircraft parameters are captured by systematic modelling of parametric and dynamic uncertainties. Both the simulation environment and the control design framework use different modelling fidelity, what are described within the paper. Reduced models are developed using two distinctive methods, respecting the control design needs: top-down balanced LPV reduction and bottom-up structure preserving methods. Based on the reduced order models various control design techniques have been elaborated by the consortium partners. In particular DLR developed and implemented a modal control method using H2 optimal blends for inputs and outputs. University of Bristol developed structured H-infinity optimal control methods, while SZTAKI proposed a worst-case gain optimal method structured controller synthesis method handling parametric and complex uncertainties. After the brief introduction of hardware-in-the-loop test setup and the description of mission scenarios the implementation issues of the baseline and flutter controllers are discussed. DLR and SZTAKI flutter controllers are evaluated in a hybrid software- / hardware-in-the-loop test setup as at this stage of development the latter can not tolerate the estimated delay of the hardware system but their comparison is advantageous before future developments. Recommendations on active flutter mitigation methods are given based on the experience of synthesis and implementation of these controllers. Flight test results will follow these experiments, once the flight testing of the flutter wing commences.

I. Introduction

FLUTTER Free FLight Envelope eXpansion for ecOnomic Performance improvement (FLEXOP), [1] is a project within the European Union's Horizon 2020 framework. Its main goal is to raise efficiency of a currently existing wing by derivative solution with higher aspect ratio at no excess structural weight. The benefit of increased span will be increased lift to drag ratio and therefore less fuel burn. The downside of the more slender wings is their higher flexibility

which leads to challenges of aero(servo)elasticity. Therefore, verification of very accurate methods for flexible mode modelling and robust flutter control synthesis are key targets within FLEXOP. One of the proposed aims of active flutter control is to fly at the same speed as the baseline aircraft of today with the enabling features of passive load alleviation and active flutter control with significantly reduced wing structural mass. The developed methods will be validated with a UAV flutter demonstrator. The 7m span, 65 kg TOW demonstrator will feature three different set of wings: One baseline rigid wing (-0), one wing designed for passive load alleviation (-2) and one very flexible wing (-1) to test active flutter control. After validating the methods and tools the potential of those technologies should be evaluated within an industry driven scale-up study. The FLEXOP demonstrator UAV is shown in Figure 1. Another research project dealing with active flutter suppression is the PAAW project in the US, [2–5].

Fig. 1 FLEXOP demonstrator aircraft

The flutter suppression control law is designed based on an appropriate control oriented model, [2, 4, 6, 7]. The linear parameter-varying (LPV) framework, [8, 9] can serve as a good approach to model ASE systems for control design since it can capture the parameter varying dynamics of the aircraft. The ASE model is based on the integration of aerodynamics, structural dynamics and flight dynamics subsystems, [10–15], (Section II.A). Ground test based model updates of the aircraft are given in Section II.B. Control oriented low order models are then obtained by two distinct approaches. The top-down balanced LPV reduction is based on [7], Section III.A and the bottom-up structure preserving method is based on the approach presented in [16], Section III.B.

Based on the reduced order models various control design techniques have been elaborated by the consortium partners, Section IV. In particular DLR developed and implemented a modal control method using H2 optimal blends for inputs and outputs. University of Bristol developed structured H-infinity optimal control methods, while SZTAKI proposed a worst-case gain optimal method structured controller synthesis method handling parametric and complex uncertainties. Flutter mitigation testing requires the use of ground testing before real flights. That's why the so-called hardware-in-the-loop (HIL) test environment is introduced, Section V. After describing the mission scenarios and the implementation issues of the controllers the delay of the HIL is measured which gives approximately the delay of the on-board hardware system. This delay is considered in baseline and flutter controller design and in all tests of the controllers. Hybrid software / hardware-in-the-loop (SIl-HIL) evaluation of the DLR and SZTAKI controllers is given in detail before HIL and real flight testing which should be the next phase of the project. Finally, the flight test aspects of the resulting solutions are briefly elaborated in Section VI.

II. Aeroelastic Modelling of a Demonstrator with Fluttering Wings

A. High fidelity nonlinear model of the FLEXOP aircraft

The ASE model of the FLEXOP aircraft is developed based on a subsystem approach as seen in Figure 2. Each of the subsystems are developed separately and combined to form the ASE model. The structural dynamics model is obtained from a detailed `Nastran` finite element (FE) model. The aerodynamics is modeled using the vortex lattice method (VLM) for steady and doublet lattice method (DLM) for unsteady models, with the provision to improve the fidelity of the aerodynamics computation using computational fluid dynamics (CFD) methods. Dynamic models for flight systems such as engines, for external disturbances, for sensors and actuators are added to form the full-order nonlinear ASE model. The nonlinear equations of motions are derived based on a mean axes reference frame, [17]. The details of the ASE model are given in [14, 15]. The resulting nonlinear ASE model of the FLEXOP aircraft consists of 12 rigid body states, 100 flexible mode states and 1040 aerodynamic lag states in addition to the actuator dynamics. This model is considered as the high-fidelity, full order model (FOM). The LPV model of such system is of too high order for control design.

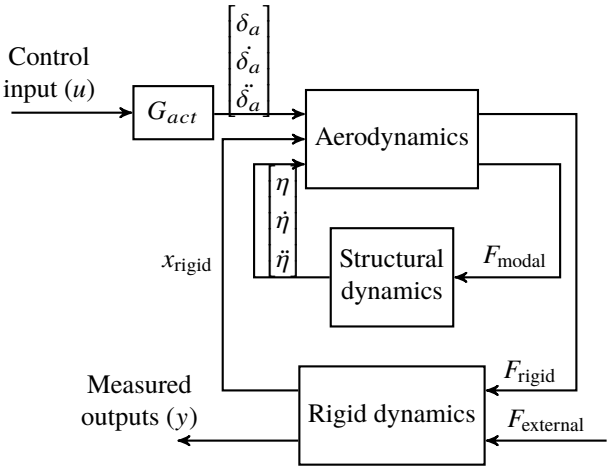

Fig. 2 ASE subsystem interconnection

In the following, updates and additions to the ASE model are presented.

B. Ground-testing based model update

A ground-test campaign [18] involving structural tests and ground vibration tests (GVT) has been performed on the FLEXOP wings. At the time of writing of this paper, a preliminary update of the FE model of the baseline wing (-0) has been performed based only on experimental data from static tests. A similar approach will be repeated also for the flutter wing (-1).

The static test was performed with the main objective being the assessment of the stiffness properties of the three wing pairs and validation of the pertinent structural models developed. Figure 3 shows the deflection of the wing-tip as a function of the applied tip-load. Shown in Figure 4 is the span-wise displacement of a wing-half subjected to 3kg load at the tip, comparing the static tests and the initial FE model. The observed difference in the stiffness could be attributed to several factors including modelling assumptions and simplifications, manufacturing deviations, material scatter, etc.

A relatively-simple first attempt at updating the stiffness model is performed by introducing a knock-down on the engineering stiffness (E_1, E_2, G_{12}) in the FE model of the wing and in the clamp used for the wing attachment. A knock-down factor of $\sim$17% in this case produces satisfactory results as shown in Figure 4. A comparison of the frequencies between this stiffness-updated FE model and the GVT [18] is shown in Table 1. Also shown is the modal assurance criterion (MAC) which is an indicator of similarity between mode shapes from two sets in Figure 5. It is seen that the FE model captures the out-of-plane bending behaviour of the wing well. On the other hand, the in-plane behaviour of the wing and the stiffness and mass modelling of the fuselage and empennage need to be investigated in more detail.

The stiffness-updated structural model serves as the basis for generating a next iteration of ASE models for controller

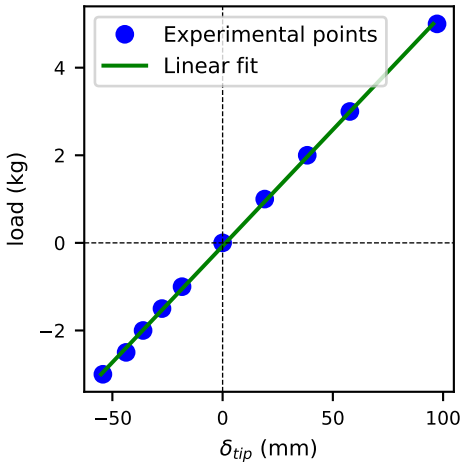

Fig. 3 Displacement vs load at tip of the wing from static tests

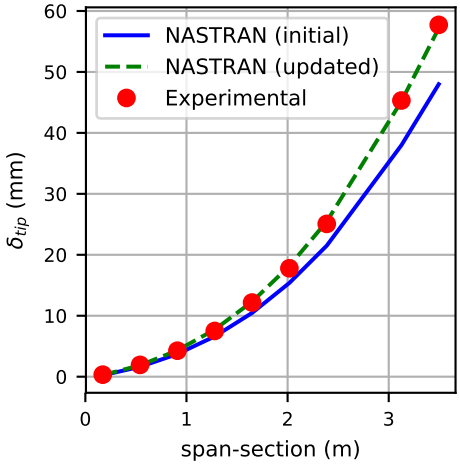

Fig. 4 Span-wise displacement of wing under tip load for the updated model

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

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

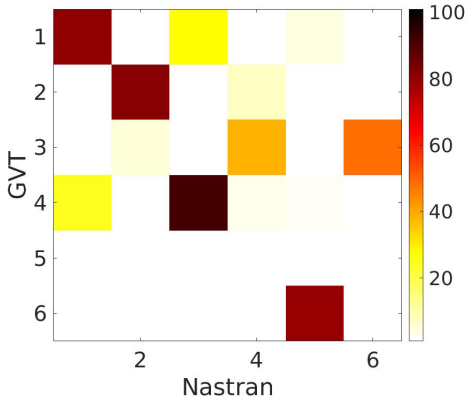

Fig. 5 MAC matrix: GVT vs stiffness-updated FE model

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

III. Model Order Reduction Methods Applied to the FLEXOP Aircraft

A. Balanced LPV reduction

A balancing based model-order reduction methodology was developed for generic parameter-dependent systems and was succesfully applied for the dynamical model of the FLEXOP demonstrator aircraft. Hereunder the main methodological steps are summerized, while more details can be found in [19]. [20] The algorithm assumes continuous time LPV systems represented by the following state-space equations:

$$G(\rho): \quad \begin{aligned} \dot{x}(t) &= A(\rho(t))x(t) + B(\rho(t))u(t) \\ y(t) &= C(\rho(t))x(t) + D(\rho(t))u(t), \end{aligned} \tag{1}$$

where, the matrix functions are given in a grid-based representation:

$$\mathcal{G} = \left\{ G_k \;\middle|\; G_k = \left[\begin{array}{c|c} cA_k & B_k \\ \hline C_k & D_k \end{array}\right], \begin{array}{l} cA_k{=}A(\rho_k),\ B_k{=}B(\rho_k), \\ C_k{=}C(\rho_k),\quad D_k{=}D(\rho_k) \end{array} \right\} \tag{2}$$

First, the parameter-dependent modal form is computed. The main idea originates from the modal form of Linear Time Invariant (LTI) plants, where the A matrix has a block-diagonal structure. In addition, the similarity transformation for modal form is constructed from the eigenvectors of the A matrix. However, to extend this structure for parameter-varying systems, one has to face at least two important problems [21]. Firstly, the consistency of the state-space, i.e. the correct ordering of the modal blocks, must be ensured over the entire parameter domain [22]. This requires the tracking of the modes between subsequent grid points. Secondly, the parameter-varying modal transformation should have a smooth parameter dependence (differentiable) in order to facilitate the smooth interpolation of the modal (and reduced) model.

Connecting the dynamical modes over the parameter domain, to ensure state-space consistency, is formulated as a minimum cost perfect matching over a bipartite graph [19].

Eigenvalues at a certain gridpoint k and at the successive one $k + 1$ are considered as two sets of vertices in a graph, where each vertex in k has exactly one pair in $k + 1$. The problem is then written as finding the correct pairing between the vertices. For this purpose, a distance metric is introduced to measure the dynamic similarity between two modes [19]. The cost of an edge in the graph then describes the dynamical similarity between the two eigenvalues on the edge. Finding the correct pairing is a minimum cost perfect matching problem [23], which can be solved very efficiently by the Hungarian Method or Kuhn-Munkres algorithm in polynomial time [24]. Applying the outlined matching algorithm over the entire parameter domain, the consistent ordering of the modal blocks are achieved.

The next step is to shape the eigenvectors to obtain a differentiable similarity transformation. This condition is formulated as a complex, unconstrained Procrustes problem as follows, where eigenspaces in neighboring grid points are rotated appropriately. The solution can be given analytically and preserves the eigen property [19]. Consequently, a parameter dependent, differentiable transformation $\bar{T}(\rho)$ can be created similarly to the LTI case.

Applying the transformation, an LPV modal form is obtained, which is state consistent and it is smoothly interpolable over the entire parameter domain. This representation is particularly useful in the reduction of large-scale systems, since:

1) Unstable or mixed stability modes (e.g., flutter modes) can be decoupled from the system and accordingly preserved in the reduced order representation. Furthermore, most of the model reduction techniques are mainly applicable for stable systems.
2) Modes outside of the frequency rage of interest can be truncated from the model. This is a very important and useful property in a control oriented reduction framework due to the presence of control bandwidth limitations.
3) The dynamical modes can be easily handled and grouped together to form subsystems with similar (reducible) dynamics. This feature will be exploited in the forthcoming section.

The third step of the algorithm is to group the modal blocks with similar dynamical properties into clusters, so that the corresponding larger dimensional subsystems can be efficiently reduced (for more details we refer again to [19]). We propose a hierarchical agglomerative clustering (HAC) framework, where the clustering is based on the eigenvalue trajectories of the LPV system. We compare two eigenvalue trajectories and characterize the similarity in terms of the dynamical response they represent. Accordingly, the comparison of two clusters is applied with the complete link clustering method: the similarity of two clusters is determined by the similarity of their most dissimilar members [25].

The result of the HAC is generally visualized by a dendrogram, which is a tree diagram illustrating how the data objects are merged into larger clusters until the one single cluster is reached. The final cluster structure is obtained by cutting the dendrogram at a user-defined level of similarity. The careful choice of this threshold is important, because it determines the number and size of the clusters generated. In the model reduction framework, this threshold is mainly determined by the available computation capabilities, i.e. the chosen model reduction methodology must be solvable for the largest cluster. In our algorithm, the balanced reduction has been chosen to reduce the dimension of the clusters.

Balanced reduction is a fundamental approach for the model reduction of linear (time invariant and varying, as well as parameter-dependent) systems [20], [26]. The key concept is the balanced realization which reveals the controllability and observability properties of the system. In balanced realization uncontrollable and unobservable states can be identified and deleted easily, without affecting the input-output behavior of the entire system.

After clustering the dynamical modes, separate LPV systems have been obtained. Then, the similarity transformations which transform the clusters into balanced form can be constructed from the corresponding observability and controllability Gramians [26]. The computation of the Gramians is carried out by solving Lyapunov inequalities [26]. This is a nonconvex optimization problem, which can be reduced to a linear optimization problem with Linear Matrix Inequality (LMI) constraints, using an iterative computational scheme (as suggested in [26]).

Having determined the observability and controllability Gramians of every subsystem, the balancing transformations and the corresponding parameter dependent, generalized singular value trajectories can be determined. The singular values characterize the controllability and observability properties of the states in the balanced realization. Therefore states with small singular values can be eliminated without affecting the IO behavior. After reducing the subsystems individually, the small dimensional subsystem dynamics are finally joined together with the unstable modes to obtain the low dimensional approximation of (2).

1. Numerical results

The algorithm was succesfully applied for the dynamical model of the FLEXOP demonstrator. The original model consists of 524 states, 38 inputs and 16 outputs given as a set of LTI systems evaluated at 26 airspeed values. The eigen-decomposition of the model is carried out first. Multiple eigenvalues, mostly related to the lag state dynamics, are grouped together. The Hungarian algorithm was applied between the grid points to connect the eigenvalue trajectories. Then the Procrustes smoothing was used for the grouped eigenspaces. Having obtained the smooth LPV modal form we were in the position to remove and store unstable or mixed stability modes. In the underlying system 3 modes have mixed stability properties (flutter and spiral mode), represented by 5 states. These states have been removed and the remaining states then reduced with modal truncation. For the suppression of the flutter phenomena a special, high bandwidth control actuator has been chosen in the demonstrator aircraft. Accordingly a 200$\frac{rad}{sec}$ bandwidth has been set for the modal truncation: faster modes have been removed. This step reduced the system to 159 states due to the large number of very fast modes.

Next, the remaining 159 state have been clustered using the proposed HAC algorithm, which actually revealed that most of the lag-state dynamics can be grouped into clusters. The controllability and observability Gramians of the smaller dimensional subsystems were computed next. Using the Gramian solutions, balancing transformation was performed and weakly observable and controllable states were truncated. The reduced systems were then merged together in a stable LPV model, which is then extended by adding back the mixed stability part. Consequently, a 35 dimensional approximation has been obtained.

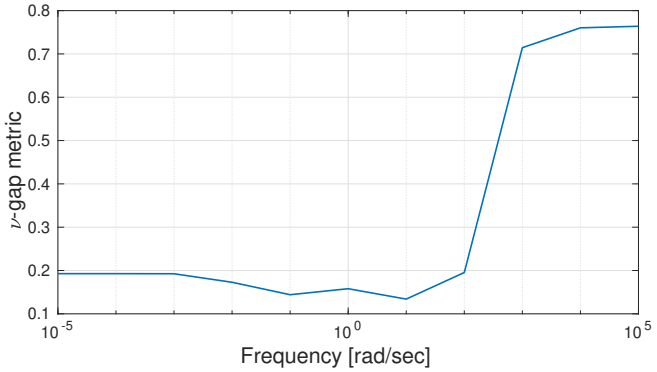

Fig. 6 Frequncy distribution of the ν-gap metric between the 524 and 35 order state models.

Finally, in order to measure the goodness of the reduced order model we adopted the ν-gap metric [27]. This metric is generally used for characterizing closeness in a closed-loop setup. Since our aim is to use the reduced order model for control design, the ν-gap metric is a suitable choice. Figure 6 shows the frequency distribution of the ν-gap distance. Here, at each frequency we have chosen the worst case value over the parameter domain Γ computed between the *interpolated* models. This is a very important feature, which has to be emphasized.

It can be seen that the distance remains reasonable low for the lower frequency domain and only increases above the prescribed frequency bandwidth used during the modal truncation.

Therefore we conclude that the reduced order approximant of the flexible aircraft can serve as a reliable basis for the control design efforts.

B. Bottom-up modeling

The bottom-up modeling is pursued in order to obtain an LPV model of the FLEXOP aircraft that is of sufficiently low order for control design. The key idea is to reduce the subsystems before the integration into the nonlinear model. The reason behind this is that the structural dynamics and aerodynamics subsystems have simpler structure than the combined ASE model. Thus, the order of these subsystems can be reduced by simpler and more tractable reduction techniques. Such approach leads to a low order ASE model (LOM).

The LPV model of the resulting LOM is compared to the LPV model of the FOM to verify its accuracy. The grid based LPV models of the LOM and FOM are derived in the following way. The nonlinear ASE model is first trimmed for straight and level flights at various airspeeds after which Jacobian linearization is carried out. The scheduling parameter is defined as $\rho = V_s$ in the interval $[30, 65]\ m/s$ over a grid of 71 equidistant points.

The ν-gap metric $\delta_\nu(\cdot,\cdot)$ is used as a measure to compare the LOM and FOM LPV models. It takes into account the feedback control objective. It takes values between zero and one, where zero is attained for two identical systems. A system P_1 that is within a distance ϵ to another system P_2 in the ν-gap metric, i. e. $\delta_\nu(P_1, P_2) < \epsilon$, will be stabilized by any feedback controller that stabilizes P_2 with a stability margin of at least ϵ, [27]. A plant at a distance greater than ϵ from the P_2, on the other hand, will in general not be stabilized by the same controller. It can be calculated frequency by frequency as

$$\delta_\nu(P_1(\mathrm{j}\,\omega), P_2(\mathrm{j}\,\omega)) = \| \left(I + P_2(\mathrm{j}\,\omega)\ P_2^*(\mathrm{j}\,\omega)\right)^{-1/2} \left(P_1(\mathrm{j}\,\omega) - P_2(\mathrm{j}\,\omega)\right) \left(I + P_1^*(\mathrm{j}\,\omega)\ P_1(\mathrm{j}\,\omega)\right)^{-1/2} \|_\infty \tag{3}$$

The ν-gap metric is a linear time invariant (LTI) technique and the goal is to evaluate it at each LPV grid point. Since the LOM is aimed for flutter suppression control design, the ν-gap metric is investigated for an input/output set that is relevant for the control design. These are $L4$, $R4$ inputs and vertical acceleration (a_z) and pitch rate (q) measurements at the c.g. and at the 12 IMUs. An additional crucial consideration to be made is to define a frequency range of interest in which it is expected that the LOM is a good approximation of the FOM. Since the goal of the control design is flutter suppression, the flutter frequency (50.2 and 45.8 rad/s) determines the frequency range for which an accurate model is required. Based on the flutter characteristics, the frequency range of interest is defined up to 100 rad/s.

1. Reduction of the structural dynamics model

The structural dynamics model is an LTI system, therefore, state truncation can be applied. Retaining the first 6 structural modes and modes 19, 20, 21 results in acceptable accuracy. This way the reduced order structural dynamics model is of 18 states as opposed to the 100 states of the full order structural dynamics model.

2. Reduction of the DLM aerodynamics

The aerodynamic lag terms can be given in the following state space form

$$\begin{aligned} \dot{x}_{aero} &= \frac{2V}{\bar{c}} A_{lag} x_{aero} + B_{lag} \begin{bmatrix} \dot{x}_{rigid} & \dot{\eta} & \dot{u} \end{bmatrix}^T \\ y_{aero} &= C_{lag} x_{aero} \end{aligned} \tag{4}$$

where V is the airspeed, $\bar{c}$ is the reference chord, $\dot{x}_{rigid}$ are the rigid body states, η represent the structural dynamics states and u is the control surface deflection. A linear balancing transformation matrix T is computed for the aerodynamics model given by A_{lag}, B_{lag} and C_{lag} in (4). The reduced order aerodynamics model is obtained by rezidualizing the states with the smallest Hankel singular values. Keeping 2 lag states results in acceptable accuracy. The ν-gap plot of the FOM and resulting LOM are shown in Figure 7.

The resulting bottom-up LOM if of 56 states, that consists of 12 rigid body states, 18 structural dynamic states, 2 aerodynamic lag states and 24 actuator dynamics states. In addition to the ν-gap plots, the pole migration, Bode plots and numerical simulation responses of the LOM and FOM are compared. Further details of the bottom-up modeling of the FLEXOP aircraft can be found in [14, 28]. Figure 8. shows the pole migrations of the LOM and FOM LPV models. The full order LPV model predicts flutter at 52 and 55 m/s at frequencies of 50.2 rad/s and 45.8 rad/s. The LOM LPV model predicts flutter at 52.5 and 56.5 m/s at frequencies of 50.3 rad/s and 46 rad/s. The flutter speed and frequency accuracy of the LOM is good enough for control design.

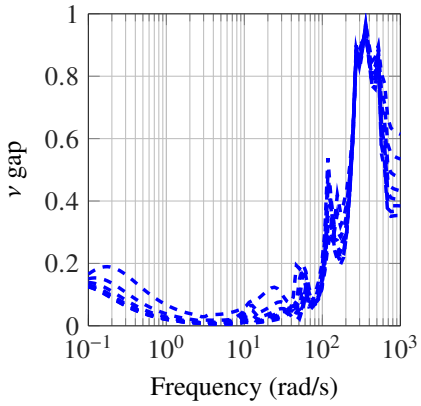

Fig. 7 ν-gap values between the FOM and LOM

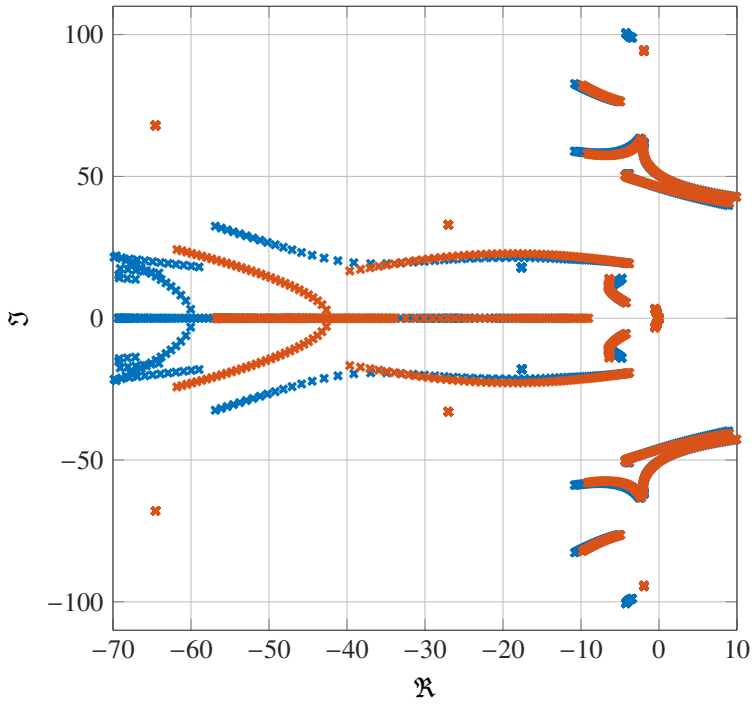

Fig. 8 Pole migration of the LOM (×) and FOM (×)

C. Uncertainty structure

The uncertainty structure is based on the derivation presented in [29] and takes the following form

$$\dot{x} = \Big[A(\rho) + D_a(\rho)\Delta_a(t)E_a(\rho)\Big]\, x + B(\rho)u \tag{5}$$

where the uncertain block $\Delta_a(t)$ satisfies

$$\|\Delta_a(t)\| \leq \frac{1}{\gamma_a}, \quad \Delta_a(t) = \Delta_a^T(t), \tag{6}$$

and $D_a(\rho)$ and $E_a(\rho)$ are known scaling matrices.

Since the aim of the control design is flutter suppression, it desirable to have robust stability in case of uncertainty in the flutter modes. 10% uncertainty is assumed in 2 elements of $A(\rho)$ that strongly influence the flutter modes. The pole migration of the flutter modes of the nominal and uncertain models are given in Figure 9.

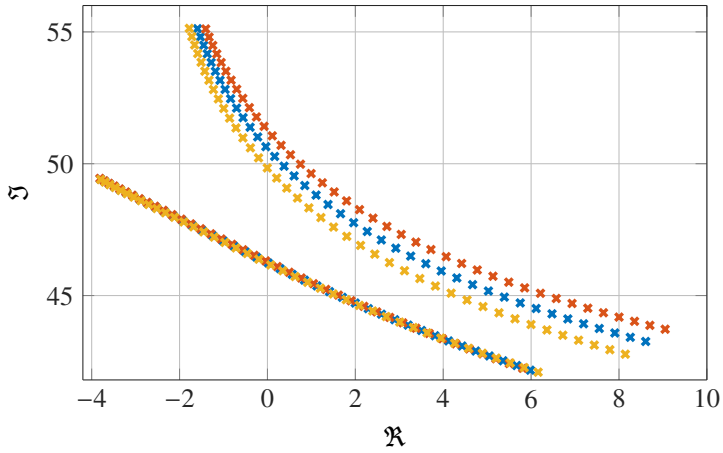

Fig. 9 Uncertainty of the flutter modes: nominal model (×), +10% uncertainty (×), -10% uncertainty (×)

IV. Control design Methods

A. H_2 optimal blending

The two flutter modes mainly limiting the operational velocity range of the aircraft can be well distinguished by its symmetric and asymmetric mode shapes. Both modes describe a dynamic coupling of the wing bending and torsion which becomes unstable above certain airspeeds. To individually stabilize the two flutter modes, the $\mathcal{H}_2$-optimal blending approach proposed in [30] is applied to the FLEXOP demonstrator. In doing so, the flutter modes are decoupled which allows for a straight forward design of two dedicated SISO control loops, one for each flutter mode.

In Figure 10, the resulting feedback interconnection is depicted, where the symmetric (SYM) and asymmetric (ASYM) flutter modes are subject to be controlled. Summarizing the input and output blending vectors in $K_u = [k_{u,\mathrm{SYM}}\ k_{u,\mathrm{ASYM}}]$ and $K_y = [k_{y,\mathrm{SYM}}\ k_{y,\mathrm{ASYM}}]$, the overall controller is

$$K(s,V) = K_u C(s,V) K_y^T,$$

where the single input and single output (SISO) controllers are collected on the diagonal of $C(s,V) = \mathrm{diag}\,(c_{\mathrm{SYM}}(s,V),\ c_{\mathrm{ASYM}}(s,V))$.

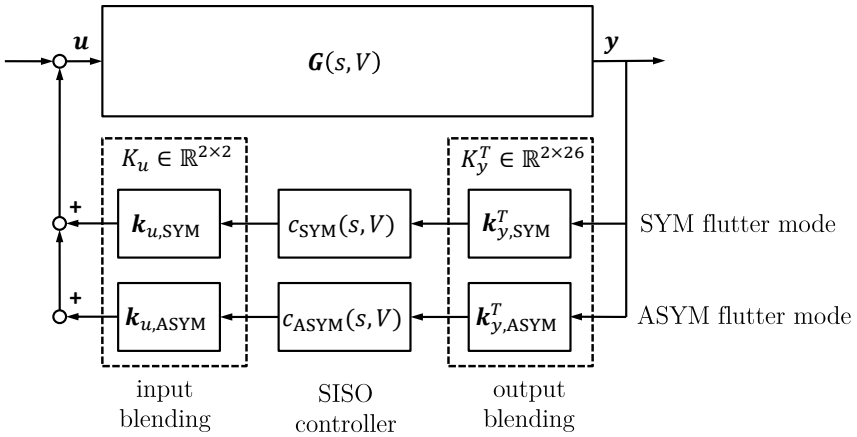

Fig. 10 Closed-loop interconnection of plant G with flutter suppression controller K, output blending matrix K_y, input blending matrix K_u, and controller C.

1. Input-Output Blending

The measurement signals considered for flutter suppression are captured by the inertia measurement unit (intertial measurement units (IMUs)) located in the wings and in the center of gravity, where only vertical acceleration and pitch rate measurements are used for the controller design herein. In Figure 11, the location of the IMUs in the wings together with the location of the ailerons, of which only the outer pair is used for flutter suppression. Before actually blending the

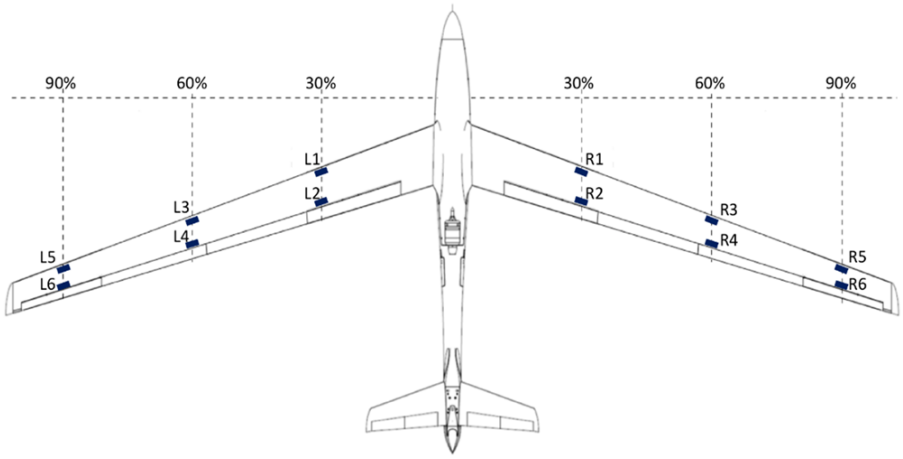

Fig. 11 Locations of the IMUs installed in the wings to measure the accelerations on the wing.

given inputs and outputs, it is proposed to normalize the rate and acceleration measurements since they are of different units, see [31] for more details. Subsequently, the $\mathcal{H}_2$-optimal blending vectors associated to the first (symmetric) and second (asymmetric) flutter mode are computed as described in [30]. The obtained input and output blending vectors basically mirror the shape of the underlying modes and hence are also symmetric and asymmetric. Since the mode shapes change only slightly within the critical airspeed range, it is sufficient to compute the blending vectors at a single airspeed $V_{\text{ias}} = 60\,\text{m}\,\text{s}^{-1}$ and hold them constant within the whole flight envelope. Applying the blending vectors, the two flutter modes are well decoupled and can be individually controlled by separate SISO controllers described as follows.

2. Single-Input Single-Output Controllers

With the derived blending vectors it is possible to design dedicated SISO controllers for the symmetric ($j = 1$) and asymmetric ($j = 2$) flutter mode. The structure of the SISO controllers is predefined as

$$c_j(V_{\text{ias}}(t)) = W_{\text{BP},j}\, W_j(V_{\text{ias}}(t)), \tag{7}$$

where $W_{\text{BP},j}$ denotes a bandpass filter to ensure that no interference with the baseline controller occurs and higher frequent modes are not excited. For both flutter modes, a second order Butterworth filter is chosen with a fixed passband from $40\,\text{rad}\,\text{s}^{-1}$ to $400\,\text{rad}\,\text{s}^{-1}$. The corresponding corner frequencies are selected such that both flutter modes are well inside the passband and controller performance is affected as little as possible. Since a large velocity range needs to be considered, the core of the flutter suppression controller $W_j(V_{\text{ias}}(t))$ is gain-scheduled. For better tuning capabilities, it is desired to keep the order of $W_j(V_{\text{ias}}(t))$ as small as possible while a larger order may allow for a better controller performance. Hence, a careful balancing between controller order and performance is required. For the first (symmetric) and second (asymmetric) flutter mode, an order of two respectively one is chosen. The state space matrices $Z_j = \{A_j, B_j, C_j, D_j\}$ of $W_j(V_{\text{ias}}(t))$ depend linearly on the indicated airspeed, i.e. $Z_j = Z_j(V_{\text{ias}}(t)) = Z_{j,0} + Z_{j,1}V_{\text{ias}}(t)$, where the matrices $Z_{j,0}$ and $Z_{j,1}$ are subject to be optimized. As explicit optimization criteria a gain margin of 6 dB and a phase margin of 45° are demanded in the optimization. The two problems are solved using non-smooth optimization techniques [32]. The resulting SISO controllers without the band-pass filter are depicted in Figure 12. Note that with increasing airspeed, the controller gain increases in the symmetric case and decreases in the asymmetric case in the frequency range of the corresponding flutter mode.

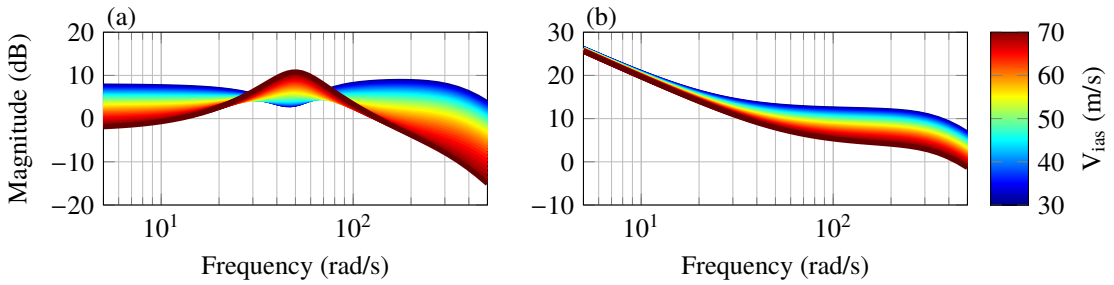

Fig. 12 Gain-scheduled SISO controllers $W_1(V_{ias}(t))$ for the symmetric mode (a) and $W_2(V_{ias}(t))$ for the asymmetric mode (b) plotted from 30 m/s to 70 m/s airspeed.

3. Linear Closed-Loop Analysis

Closing the two SISO loops stabilizes the two flutter modes as it is illustrated in the pole migration plot in Figure 13. The plot compares the open-loop poles in gray to the closed-loop poles depicted in color in dependence of the airspeed. Clearly visible is the unstable behavior, i.e., the crossing to the right half plain of the first (symmetric) and second (asymmetric) flutter mode in the open-loop. With the flutter suppression controller the symmetric flutter mode can be stabilized up to airspeeds of 65.5 m/s. The asymmetric mode is stabilized even beyond 70 m/s. Demanding additional single-loop robustness margins of 6 dB in gain and 45° in phase to the critical point, leads to a maximum operational speed of about 60 m/s. This still results in an increase in allowable speed of more than 15 % compared to the case without active flutter suppression. Also noticeable is that the other poles of the system(s) are not largely affected by the flutter suppression controller. This is acceptable since damping is rather increased than decreased.

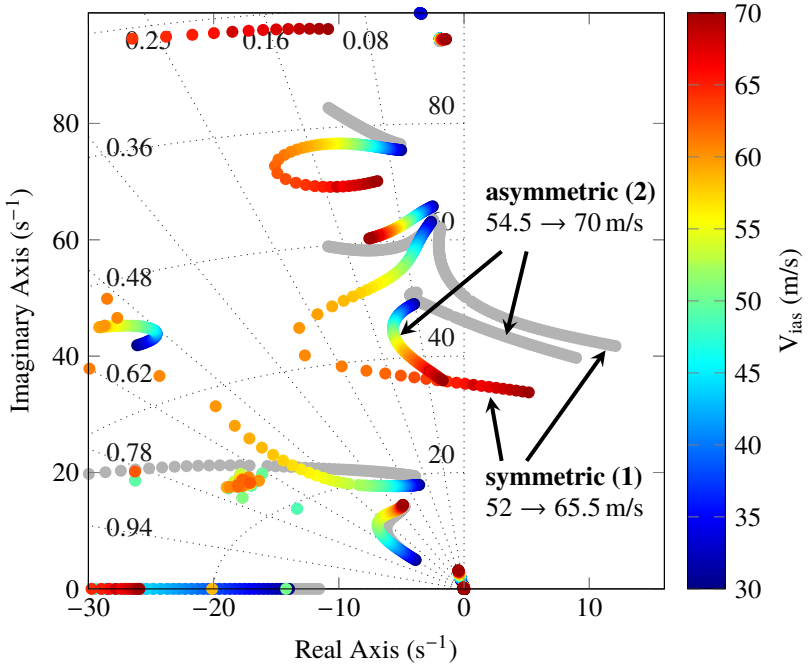

Fig. 13 Comparison of the open-loop poles in gray and closed-loop poles (colored) in dependence of the indicated airspeed V_{ias}. Only the positive imaginary axis is depicted for readability reasons.

B. Structured H_∞ method

1. Standard and Structured H_∞ flutter control

In this section, the control design activities by UBRISTOL are summarized. They are presented in more detail in references [7, 33, 34] to which the reader is referred.

The control design theory used is H_∞optimization but following two distinct algorithms: standard and structured [35] (the former is subsequently referred to as non-structured in order to differentiate it from the latter). These approaches were selected as they have already been used by teams in the US for a similar active flutter control activity [6] and because UBRISTOL team members had a long heritage on using the techniques [36–38], including manned flight tests [39].

Four main design steps were followed by UBRISTOL in order to ensure understanding of the active flutter control problem and the incremental maturation of the design. Each is briefly described in the following subsections.

2. Knowledge acquisition step

For this step the work from [6] was used to provide an initial evaluation on the appropriateness of the design approaches. Although, as opposed to FLEXOP, this reference looked at the Body Freedom Flutter (BFF) problem and a much scaled down UAV, it was fundamental for the work presented in here. In a first step, the same plant and controller design objectives/weights/interconnection were used but instead of using the non-structured H_∞ as in that reference, the structured approach was used. The results indicated direct improvements on the robustness and enlargement of the flight envelop by the resulting structured H_∞ controllers (despite using exactly the same weights and interconnection).

3. Preliminary (coupled standard and coupled structured) H_∞ LTI design step

This step served to pose the H_∞ optimization problem for the FLEXOP demonstrator (i.e. for the 1152 states high-fidelity models obtained by DLR). Reference [7] provides a more detailed presentation. The two algorithms were applied to a single LTI plant corresponding to an airspeed of 57 m/s (N=13) and of order 38 (reduced from the 1152 via balanced and Hankel order reduction methods), augmented to 48 after inclusion of delays, actuators and sensors (captured in two equal 5-states model G_{red}). It is important to note that by coupled it is meant that the controller was designed to tackle simultaneously the symmetric and antisymmetric flutter modes. This will change for the next step. The controller had two outputs (L4 and R4 ailerons) and 4 inputs (pitch rate q, and normal accelerations at the center of gravity and the left and right wings). The same weights and interconnection, see Figure 14, were used for both H_∞ approaches with the main difference that the non-structured approach yielded a controller of order 48 which was then reduced to order 4, while the structured approach used this controller as the initial optimization guess and fixed the order directly to 4 states. The results were promising for both techniques, although the structured controller had a lower range of validity (in terms of airspeed coverage).

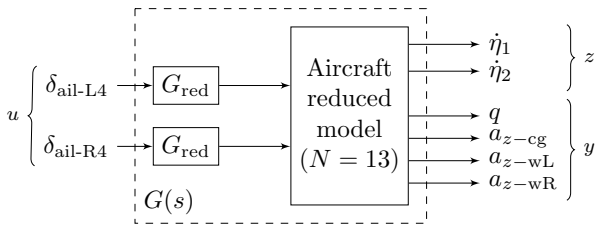

Fig. 14 Preliminary H_∞ flutter design: synthesis plant [7]

4. Consolidated (coupled standard and decoupled structured) H_∞ LTI design step

As updated high-fidelity models of the FLEXOP demonstrator were released by DLR, the design teams had to verify (and when necessary update) their controller designs. UBRISTOL took advantage of the time in between these model releases and explored alternative control architectures, i.e. different design interconnections (input/outputs) with different weights and different plants. A more detailed presentation of the results for this step is given in reference [33] but as a summary:

- The design plant was now of order 40 states (plus the addition of the extra 10 states from the actuator-sensor-delay chain G_{red}) and obtained at an airspeed of 59 m/s (N=15). Furthermore, as seen in Figure 15, the input and outputs of the synthesis plant (and thus of the obtained controllers) were different with respect to those in the preliminary approach. In this case, the pitch rate was removed as a plant output and the controller input/outputs were generalized to symmetric (subscript S) and antisymmetric (subscript AS) channels.

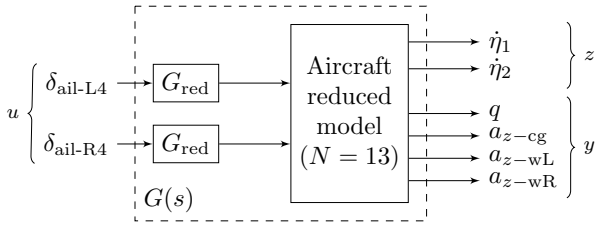

Fig. 15 Consolidated H_∞ flutter design: synthesis plant [33]

- Two different design approaches were used: (C.1) a coupled non-structured approach following the preliminary one but with different controller inputs/outputs and weights (i.e. different values and also reduced dimensions due to the smaller dimension of the channel y); and (C.2) a decoupled structured approach using the same synthesis plant but with the controller structured to be de-coupled, see Figure 16. Note that each of the two decoupled controller components K_S and K_{AS} have different dimensions, specifically order 9 and 6 states respectively –these dimensions were chosen based on the controller reduction performed for the C.1 controller which yielded acceptable performance when reduced from 50 to 15 states.

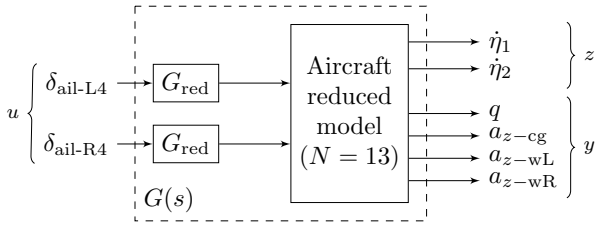

Fig. 16 Consolidated H_∞ flutter design: coupled (left) versus decoupled (right) controllers [33]

The results indicated improved robustness and flight envelop coverage, with respect to the preliminary designs. Further, it was also noted that the structured controller design was overly sensitive (at least for the considered plant and posed optimization problem) to the initial guess so for the final design step the focus was on using the coupled non-structured approach.

5. Final scheduled coupled standard H_∞ design step

For the final design, see reference [34] for details, three LTI controllers were obtained using the same synthesis plant interconnection as in Figure 15 but for three $59, 63, 66$ m/sec (plant numbers $N = 15, 19, 22$ respectively). In addition to the change of plants used for the LTI design, which already resulted in different values for the weights, a disturbance weight that was used before during the consolidated interconnection synthesis was now removed in order to reduce the overall complexity of the tuning process. The scheduled controller was obtained by interpolation of the outputs of the three reduced LTI controllers using a piecewise-linear scheduling rule based on true airspeed V_{TAS} (although the indicated airspeed was also used), see Figure 17. This output interpolation was preferred as the reduced controllers had different state dimension (respectively 17, 16 and 15 states).

The verification and validation results (including gain/phase margin analyses, and time simulations without and with the baseline controller) presented in reference [34] show that the final scheduled flutter controller is capable of extending stability beyond the open-loop flutter speed of V_{TAS}= 52 m/s up to V_{TAS}= 62.4 m/s, which is an improvement of about 22%, in the presence of the baseline controller (and up toV_{TAS}= 69 m/s when tested without it).

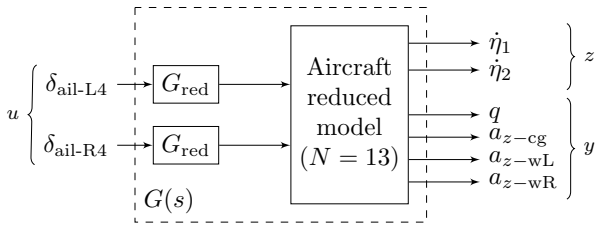

Fig. 17 Final scheduled H_∞ flutter design: interpolation rule [R3]

C. Worst-case gain optimal structured controller synthesis

The structured control design technique used for flutter suppression control is detailed in [40]. It's MATLAB implementation is available at [41]. The key idea of the algorithm is repeated here for ease of reference. The content of the present section is expanded in [42].

The general closed loop interconnection is depicted in Fig. 18. Here, $M(s)$ is a Linear Time Invariant (LTI) system (usually called the generalized plant). The uncertainty block $\Delta(s)$ is stable and structured with $\|\Delta(s)\|_\infty \leq 1$. Both dynamic and parametric uncertainty appear in $\Delta(s)$. The uncertain plant is denoted by $P(\Delta)$. The structured controller $K(\kappa)$ has fixed dynamics with tunable parameters in the vector κ. The worst-case gain of the closed loop in Fig. 18 is the maximal H_∞ norm from the input d to the output e over the allowable set of uncertainty. The objective of the control design is to tune κ to minimize the worst-case gain of the closed loop.

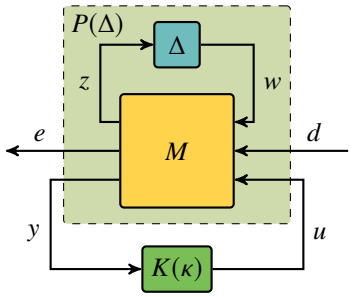

Fig. 18 Closed loop interconnection for the worst-case gain optimal control design.

The synthesis algorithm in [40] provides a solution to this problem. It collects worst-case samples of the uncertain parameters and synthesizes unique D-scales for the individual samples using a convex optimization method. The structured controller is tuned for the collection of scaled samples simultaneously. To apply this method, the flutter suppression problem is articulated as the robust stabilization of the uncertain aircraft model. The construction of the uncertain plant is detailed in the following Section. In the next Section after that, the control design setup and the result of the synthesis are given.

1. Uncertainty Modeling

The control input for flutter suppression is the aileron deflection command received by the actuators of the outermost ailerons on the wing denoted by $\delta_{\text{a,L}}$ and $\delta_{\text{a,R}}$ for the left and right wing respectively. The three measured signals are angular rates along the horizontal axis of the aircraft: $r_{y,\text{L}}$, $r_{y,\text{R}}$, and q. The signal q is measured in the center of gravity. On the left and right wing respectively, $r_{y,\text{L}}$ and $r_{y,\text{R}}$ are measured at 90% of the length of the wing close to the trailing edge. Using the combination of these measured signals depicted in Fig. 19, the symmetric and asymmetric flutter modes are isolated by the creation of two SISO plants and two SISO control loops for their stabilization. The states corresponding to the longitudinal motion and to the symmetric deformations are removed from the system representing the asymmetric flutter mode and vice versa.

As the results of the bottom up modeling and reduction technique, the low order SISO models are given on a three dimensional grid. In the grid, there are 36 equidistant points of the airspeed between 30 and 65 m/s. There are also five points of the natural frequency and damping of the structural dynamics between ±1% and ±10% of their nominal value

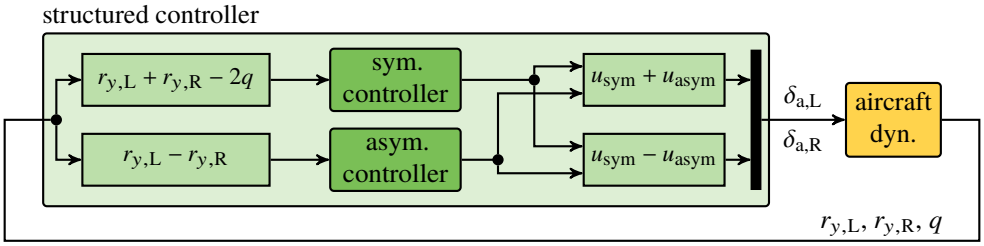

Fig. 19 Structure of the control loop using two SISO controllers to stabilize the symmetric and asymmetric flutter modes separately.

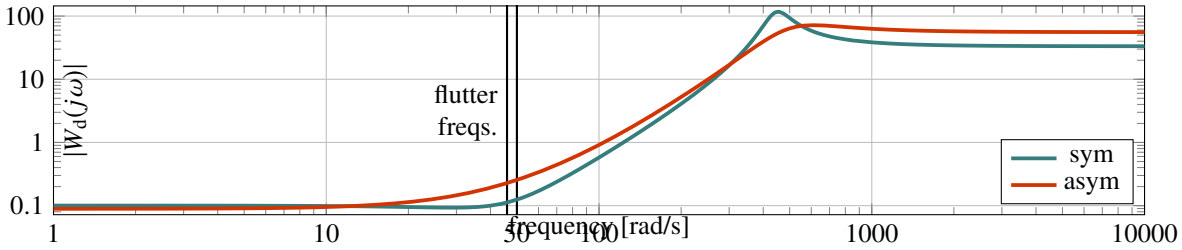

Fig. 20 Bode magnitude plot of the dynamic uncertainty weights.

respectively. Least squares fit is applied to get the uncertain state-space matrices of the systems in the form

$$
\begin{aligned}
A_\delta &= A_0 + A_1\delta_v + A_2\delta_v^2 + A_3\delta_{\omega_0} + A_4\delta_{\omega_0}^2 + A_5\delta_\xi \\
B_\delta &= B_0 + B_1\delta_v + B_2\delta_v^2 \\
C_\delta &= C_0 + C_1\delta_v + C_2\delta_v^2 \\
D_\delta &= 0,
\end{aligned}
$$

where $|\delta_v| \leq 1$, $\left|\delta_{\omega_0}\right| \leq 1$, and $\left|\delta_\xi\right| \leq 1$. Dynamic uncertainty is added to account for the neglected dynamics. The form of the uncertain model is then $G(s) = C_\delta\,(sI - A_\delta)^{-1}\,B_\delta\,(1 + W_{\mathrm{d}}(s)\,\Delta_{\mathrm{d}}(s))$, where $\Delta_{\mathrm{d}}(s)$ is the stable SISO dynamic uncertainty with $\|\Delta_{\mathrm{d}}(s)\|_\infty \leq 1$, and $W_{\mathrm{d}}(s)$ is the weight of the dynamic uncertainty. Since the uncertainty model is identical for the two systems, $G(s)$ may refer to both the symmetric of the asymmetric model. The weights are chosen so that the uncertainty is low on low frequency up to the flutter frequencies and rises to hight levels on high frequencies as illustrated in Fig. 20.

2. Controller synthesis

The generalized plant interconnection for both the symmetric and asymmetric loop is depicted in Fig. 21. The objective of the design is robust stabilization with acceptable control effort. To this end, the sensitivity function is weighted by $W_S(s) = \frac{1}{2}$ and the control input is weighted by $W_u(s) = \frac{1}{10^\circ} = 5.78$. The fourth order Padé approximation of 15 ms output delay is added to the system denoted by $D(s)$. The plant $G(s)$ in Fig. 21 is again either of the two uncertain SISO systems described previously. The filter $F(s)$ is used to enforce the desired bandwidth of the controller. The bandwidth must agree with the limitation posed by the actuator and the sampling frequency of the hardware in which the controller is to be implemented (200 Hz). Also, minimizing the sensitivity function must not come at the price of exciting high frequency and therefore necessarily uncertain dynamics as pointed out by [43]. The Bode magnitude plot of the filter satisfying these criteria is depicted in Fig. 22.

The resulting controller is analyzed in closed loop with the high fidelity model and the baseline controller. As shown in Fig. 23, the controller stabilizes the (nominal) system up to 68 m/s. It increases the damping of both flutter modes significantly. In order to assess the robustness of the closed loop, loop-at-a-time stability margins are calculated. The results are depicted in Fig. 24. (Note that the margins for the channel q are higher than for the rest of the channels. This due to the fact that q is subject to both the flutter and the baseline control laws.) The gain and phase margins of all channels are deemed acceptable up to 60 m/s which is therefore nominated as the robust flutter speed. The margins are

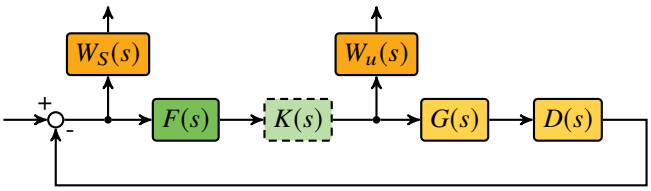

Fig. 21 Generalized plant interconnection.

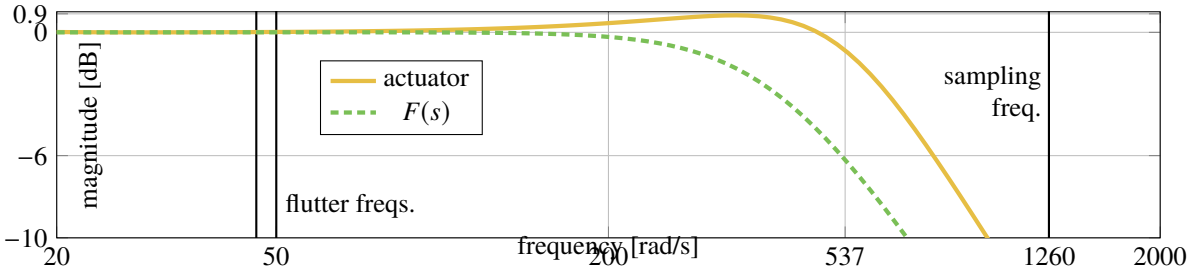

Fig. 22 Bode magnitude plot of $F(s)$ and the performance constraints.

degrading gracefully beyond this speed and become zero at the absolute flutter speed which is 68 m/s. These results indicate that the aircraft is safe to fly up to 60 m/s which means that the flight envelope was extended by 15%.

V. Hardware-in-the-Loop Testing of the Control Methods

After design and development its important to test the autopilot code in safe circumstances before doing real flight testing. Basically two test setups were created to do this. The first is the so-called Software-in-the-loop (SIL) simulation where the aircraft (A/C) model the baseline and flutter controllers are all run in Matlab / Simulink completely excluding FCC hardware. The second is the so-called Hardware-in-the-loop (HIL) setup where all control algorithms run on the FCC hardware and only the aircraft is simulated in Matlab / Simulink. This is the last stage which should be passed before real flight testing. A detailed scheme of the HIL simulation is shown in Fig. 25 while its photo is shown in Fig. 26. Fig. 25 shows that the complete HIL setup consists of a Matlab / Simulink model of the aircraft together with actuator models and emulation of sensors completed with visualization and the real hardware part (the same as applied on-board the A/C). In our setup visualization is done by Flightgear software fed with position and orientation data from Matlab. The hardware part is the FCC with Raspberry PI computer, FlightHAT (interface board between sensors, PI and RxMux) and RxMux which is the multiplexer of manual and autopilot commands. Integral parts of the hardware are the RC transmitter (Remote control) and receiver and the telemetry data link together with ground control station.

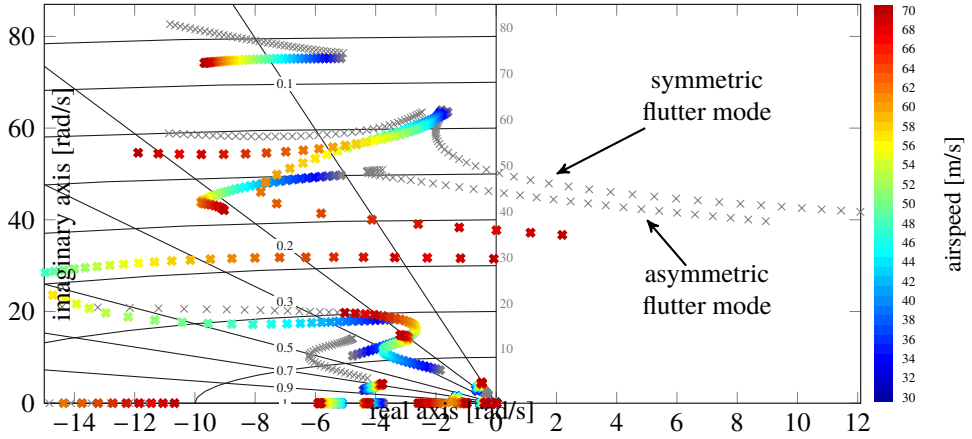

Fig. 23 Change in the pole trajectories due to the flutter controller. (The open-loop pole trajectories are depicted in gray.)

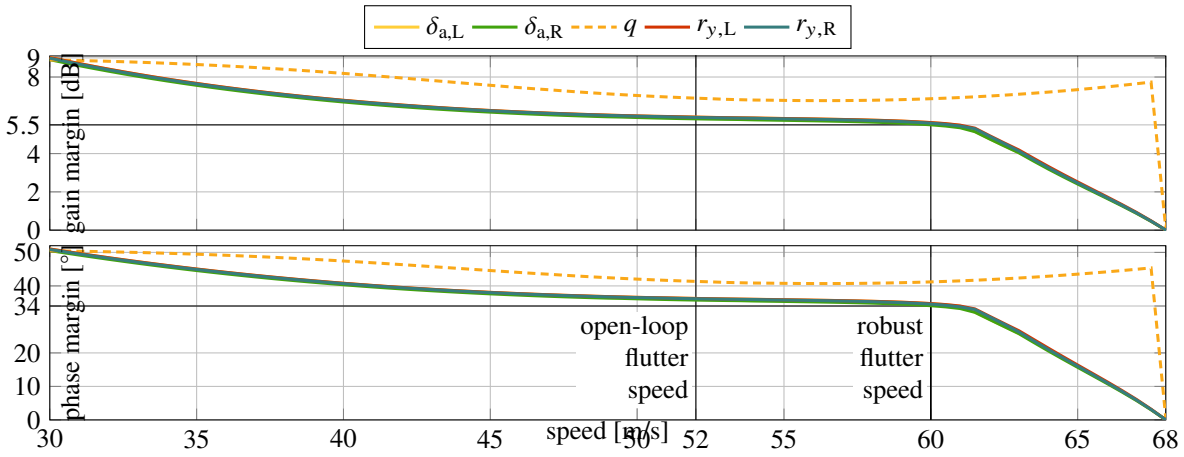

Fig. 24 Loop-at-a-time stability margins of the closed loop.

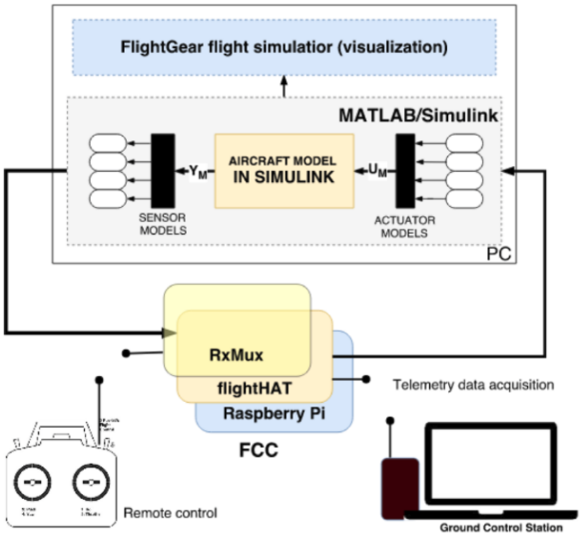

Fig. 25 Overview of HIL simulation setup

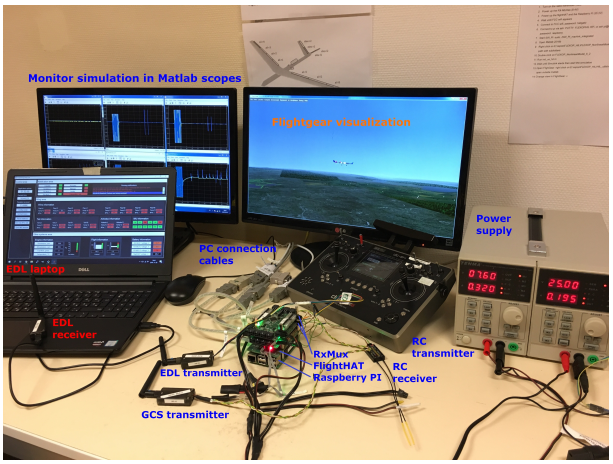

Fig. 26 Picture of the HIL simulation setup at SZTAKI

A. Mission scenario definition

The original mission scenario for the test of the flutter controllers is a racetrack pattern including two straight segments and two half turns as shown in Fig. 27. This was demanded by the limited airspace available for testing. One of the straight segments can be aligned with actual wind direction providing a headwind scenario for flutter test to have lower ground relative speed and so more time for the test. The flutter controllers can be tested by gradually increasing the airspeed on this part of the track after every loop. However, in HIL test there is no limited airspace so its faster and easier to fly straight ahead and increase the airspeed in given steps. This has led to the flight scenario with constant (800m AMS) altitude hold with constant course angle hold and a staircase IAS reference starting from 45 m/s and going up to 65-66m/s in 1m/s steps every 15 seconds (see Fig. 30 for example). The HIL simulation should be stopped if the IAS suddenly decreases because of flutter of the wings.

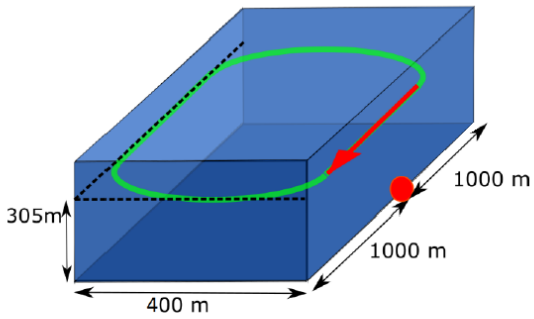

Fig. 27 The designed racetrack pattern for flutter tests

B. Controller implementation aspects

Before conducting HIL tests, implementation of the controllers (baseline and flutter) on the real FCC hardware should be performed. All of the controllers are constructed in Matlab / Simulink as discrete time models with 5ms sampling and the executable code is built from this after SIL test runs prove the correct functionality of the controllers. The baseline controller implements indicated airspeed (IAS) and altitude tracking together with the tracking of the racetrack pattern. This controller consists of simple PID and gain-scheduled PID loops so its building into executable is straightforward. Considering the flutter controllers the SZTAKI version has a small, time invariant state space model with 10 states, 2 inputs and 2 outputs which is easy to implement and run real time. On the other hand the DLR version has a linear parameter varying (LPV) state space model with 11 states, 26 inputs and 2 outputs. It is linearly interpolated between different aircraft airspeeds. As real time execution requires the reduction of the computational needs its worth to implement the linear interpolation of matrices in closed form. The two corner models of the parameter space for $30m/s$ and $70m/s$ IAS are the results of design as these are enough to cover the whole space because of linear interpolation. Denoting the related state space models as A_{30}, B_{30}, C_{30} and A_{70}, B_{70}, C_{70} (the D matrices are all zero) one can reformulate interpolation as follows:

$$
\begin{aligned}
A(V) &= A_{30} + \frac{A_{70} - A_{30}}{70 - 30}(V - 30) \\
A(V) &= \underbrace{A_{30} - 30\frac{A_{70} - A_{30}}{70 - 30}}_{A0} + \underbrace{\frac{A_{70} - A_{30}}{70 - 30}}_{AV} V \\
A(V) &= A0 + AV \cdot V \\
B(V) &= B0 + BV \cdot V \\
C(V) &= C0 + CV \cdot V
\end{aligned}
\tag{8}
$$

where V is the actual airspeed and $B0, C0, BV, CV$ are formulated similarly to the components of the $A(V)$ matrix. These matrices can be a priori calculated and so well applied in controller implementation. After doing the implementation the next step is the HIL test of controllers.

C. Control performance comparison on high-fidelity models

Before designing the baseline and flutter controllers and doing the test runs in HIL the delay of the HIL loop was measured. It was tested by sending a square signal to a very simple controller and receiving it back. The time shift between the two square signals shows the approximate time delay of the hardware part. Its only approximate because includes the delay of Matlab simulation and interface hardware cards in the PC also. The result of delay measurement is shown in Fig. 28. The measured approximate delay is 20ms including the one time step delay of the discrete time controller which is 5ms. As the resolution of measurement is 5ms (sampling of controller) one can state that

the hardware delay is about 15ms. This delay was considered in the design of the baseline (see [44]) and the flutter controllers. DLR flutter controller was designed for 4.0 and 5.0 models as presented in [44]. SZTAKI flutter controller was designed for the 4.0 model version of the aircraft provided by DLR. The design method is presented in [45] and detailed in [46] (in hungarian).

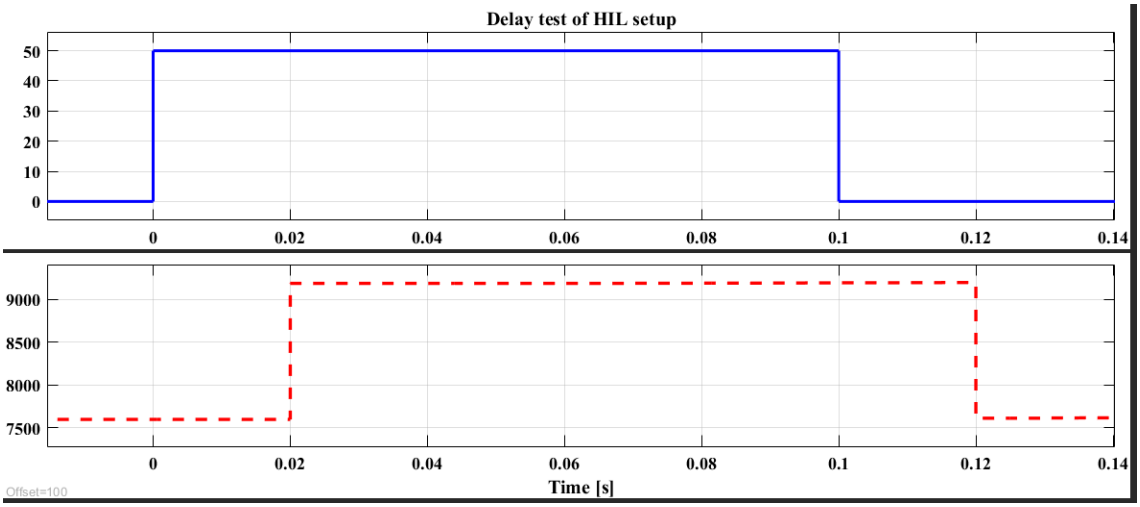

Fig. 28 Delay measurement in HIL. Continuous line is the signal out, dashed line is the echoed signal from FCC

The first step of flutter controller (together with baseline) test in HIL was to verify control performance in SIL as this is the ideal environment with user definable delay and other properties. Real-flight test results with the baseline controller has shown that only the 5.0 model version of the aircraft fits the real flight results so this should be used in any further SIL / HIL tests. Unfortunately, test of the flutter controllers in SIL on 5.0 aircraft (A/C) model has shown that the performance of the SZTAKI controller was unsatisfactory with 14ms delay and above, so it should be further developed before HIL and flight testing (more details about SIL tests can be found in see [47]). This has led to the definition of a hybrid SIL-HIL test environment (see Fig. 29) where the baseline controller is run on the FCC hardware while the flutter control is implemented in Matlab / Simulink software. This way the delay of the flutter controller remains configurable, so pre-verification of the designed controllers can be done before redesigning the SZTAKI controller for the 5.0 model. This is the focus of the rest of this chapter below.

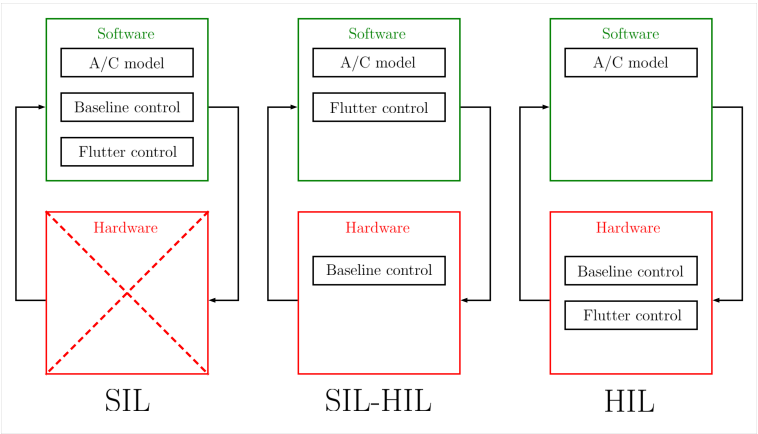

Fig. 29 Three different autopilot ground test configurations

Both DLR and SZTAKI flutter controllers were run in SIL-HIL applying different loop delays and running the baseline controller on the real hardware. An open loop simulation (OLP) with only baseline controller and then both

flutter controllers with 13, 14 and 15ms delays were run to evaluate performance. The limit speeds and flutter caused fail are summarized in Table 2. Fail means sudden decrease of aircraft airspeed because of flutter, pass means that the airspeed was held for 50-100s time so flutter compensation is successful. Note that 64.5 m/s is the maximum achievable IAS of the simulated A/C model. Which is interesting is that the DLR controller performed better with higher delay than with lower however, it was designed for 15ms delay so too low delay can also cause performance degradation. The IAS tracking results are shown in Fig.s 30 to 36. SZTAKI flutter controller performance decreased with increasing delay as expected showing only slight improvement relative to the open loop flutter speed with 15ms delay (see the Table). That's why redesign is required in the future.

Table 2 Flutter speed results

Delay	13 ms		14 ms		15 ms	
OLP	50 m/s	fail	N/A	N/A	N/A	N/A
DLR	64 m/s	fail	64.5 m/s	pass	64.5 m/s	pass
SZTAKI	63 m/s	fail	60 m/s	fail	54 m/s	fail

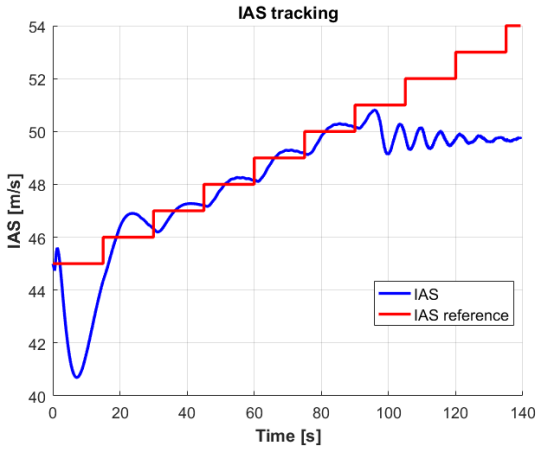

Fig. 30 Baseline open loop staircase IAS tracking

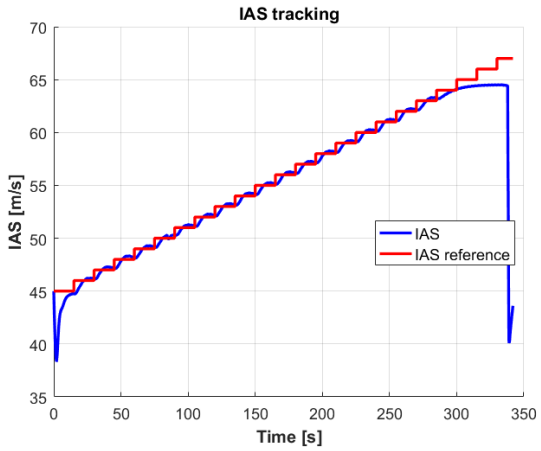

Fig. 31 DLR closed loop staircase IAS tracking with 13ms delay

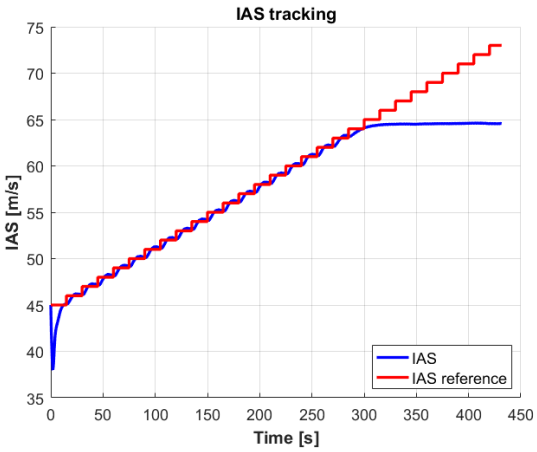

Fig. 32 DLR closed loop staircase IAS tracking with 14ms delay

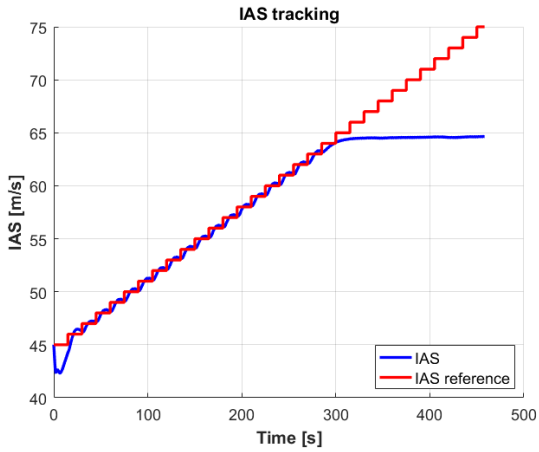

Fig. 33 DLR closed loop staircase IAS tracking with 15ms delay

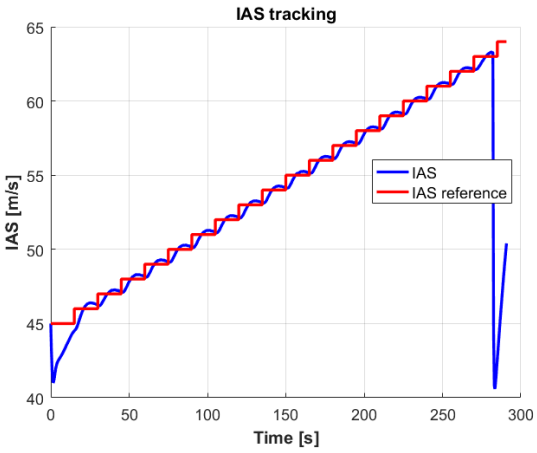

Fig. 34 SZTAKI closed loop staircase IAS tracking with 13ms delay

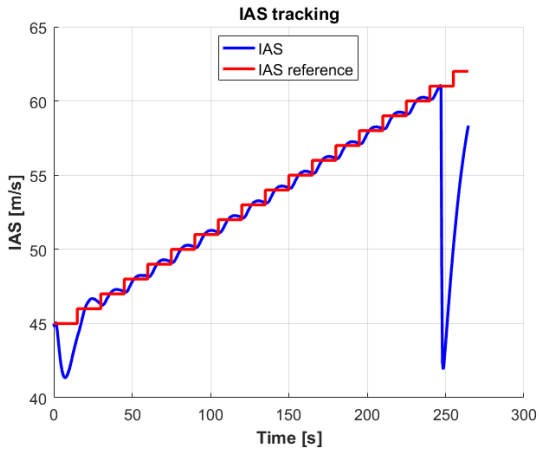

Fig. 35 SZTAKI closed loop staircase IAS tracking with 14ms delay

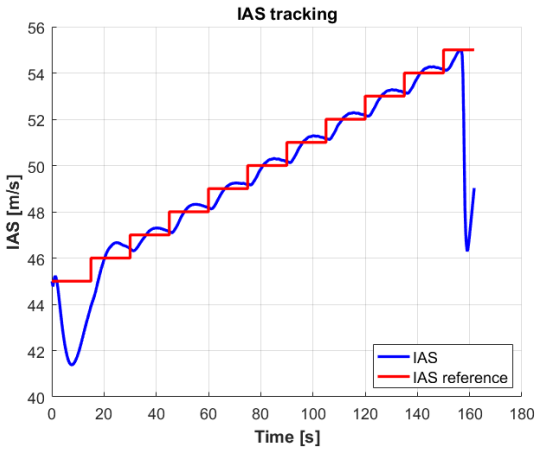

Fig. 36 SZTAKI closed loop staircase IAS tracking with 15ms delay

Detailed control input and flutter mode energy results are collected in the Appendix A. Fig.s 41 to 43 show the baseline and flutter controller input energies for the different controllers and delays. Energy here means the squares of the signals summarized for a given time interval. As the open loop flutter airspeed is about 50m/s and the maximum airspeed is 64.5 m/s the energy calculations are done from 51 to 65m/s reference values summing up the squared signals separately for every constant reference IAS section. That's why IAS is on the horizontal axis of all the figures. Fig. 43 shows that the ruddervator energy increases as the airspeed increases and this is mainly because of the modification of elevator trim value. In case of the DLR flutter controller only the trend of the ruddervator energy increases there are airspeeds with more energy and others with less. In case of the SZTAKI flutter controller the trend is linearly increasing. This shows that the DLR flutter controller has some coupling (some effect on) with the ruddervators while the SZTAKI controller does not. The sections going down in case of SZTAKI control are caused by the stop of the simulation after flutter control fails. From this point there is no data to sum up and this is the same in all other figures where a sudden drop to almost zero value can be seen.

Fig. 41 shows that with the DLR flutter controller the baseline controller uses the same amount of control for any delay and the deflections are moderate (also for aileron 3 and aileron 1). On the other hand with the SZTAKI flutter controller also the baseline controller uses a large amount of control in aileron deflections. This can be seen in the much larger energy values in the figure. The situation is the same regarding the flutter control part (Aileron 4 in Fig. 42) where the DLR controller uses moderate deflections while the SZTAKI controller applies much larger control energy. The figure also shows that the DLR control input decreases as the system delay approaches the value considered in flutter control design and does not have a well defined trend in energy changes. It can both increase or decrease by the increase of airspeed. The same was seen in the ruddervator figure.

The larger control energy input of SZTAKI flutter controller can be advantageous if it gives better damping for the flexible dynamics of wings. To check this the measured accelerations and angular rates at the wingtip (IMU6) are processed together with the modal velocities of the first, third and fifth modal coordinates (which are the first three most excited coordinates). These are all proportional with the stored energy in the wings. In case of acceleration and angular rate signals the body accelerations and angular rates are first subtracted to get relative values and these are squared and summed as in case of the control deflections. Fig.s 44 and 45 show that the SZTAKI flutter controller allows much larger accelerations and angular rates when the wing approaches flutter. This is also verified with the modal energy terms (see Fig.s 46 to 48) where the stored energy in the wings is much larger with SZTAKI flutter control than with DLR. In these figures the squared modal velocities are summed in the same manner as the control deflections.

Summarizing the SIL-HIL test results the test shows that the baseline controller works well until wing flutter occurs. Above the 50-51 m/s flutter speed the baseline controller is incapable to stabilize the aircraft. Comparing the DLR and SZTAKI flutter controllers has led to the result that the DLR controller is capable to stabilize the wing even until the maximum airspeed of the aircraft if the delay of the system matches the delay considered in control design so HIL and real flight testing of this controller can be started to be prepared. On the other hand the SZTAKI controller uses much more control energy and induces much more energy in the flexible dynamics and is unable to stabilize the wing on the whole airspeed range. By the increase of system delay its capabilities become even more limited so it should be redesigned and improved before HIL and real flight tests.

VI. Flight Testing Aspects of Active Flutter Mitigation

A. Actuator Limitations

Based on the model results, the approximated flutter frequency is round $8Hz$. The expected actuator load is at a maximum of $4Nm$. An off the shelf actuators, the maximum load is usually a given parameter by the manufacturer, but usually, there is no available information about the bandwidth itself. To be able to safely control the flutter, the actuator needs at least double the bandwidth than the flutter frequency itself.

Measurement was made on an off the shelf MKS HBL 599 actuator, which has enough speed and torque characteristics to serve as an actuator in this case. The measurements showed 37, that the actuator itself have bandwidth around $12 - 14Hz$. This available bandwidth is just not sufficient in our case - not even counting with safety margins.

To face the bandwidth requirement, a custom direct drive solution was proposed. A light-weight BLDC motor with high-resolution encoder and a sufficient drive electronics was put together for testing 38. The main components of the custom actuator are a T-Morot U10PLUS KV80 actuator, Elmo, Gold Solo Twitter servo drive and a Netzer DL-25 absolute position rotary encoder.

The initial measurements showed, that this custom solution has high enough bandwidth even with safety margins to

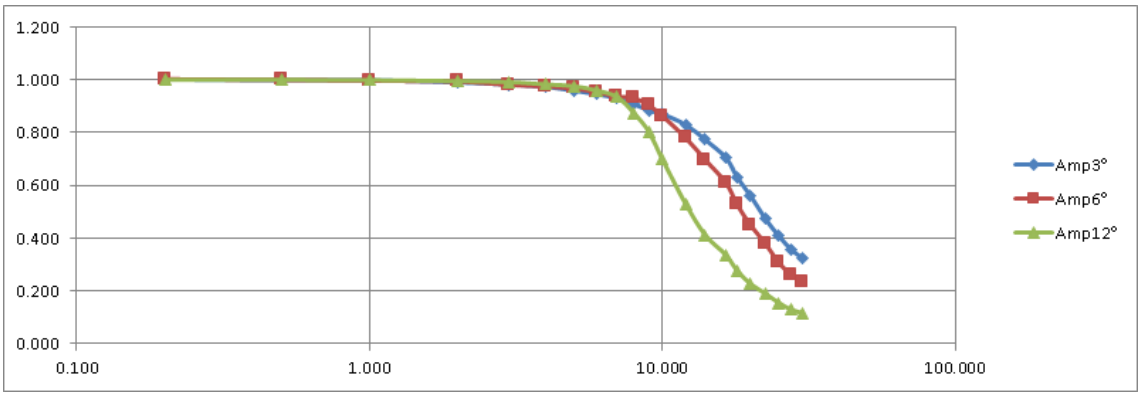

Fig. 37 Amplitude characteristics of the HBL 599 servo with $2.4 * 10^{-4} kgm^2$ inertial load

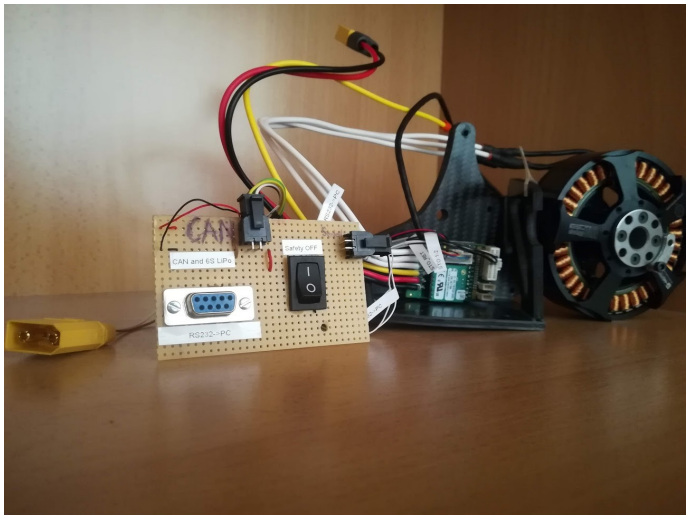

Fig. 38 Assembled custom direct drive servo solution for active flutter suppression

be used for active flutter suppression. With the given drive electronics, the fine-tuning of the actuator parameters are possible 39.

Another interesting feature on the custom direct-drive system is the step response of the actuator itself 40. The setting time itself is much faster than in the standard geared servo like the HBL 599.

VII. Conclusion

The paper presented modeling aspects of aeroservoelastic systems including finite element modeling updates based on ground tests. Two different directions, the top-down and bottom-up modeling approaches were used to obtain low order control oriented LPV models. Both reduced similar number of states and similar approximations of the high fidelity nonlinear aeroservoelastic model of the FLEXOP demonstrator aircraft. The reduced order models were augmented by parametric and dynamic uncertainties. These uncertain LPV models were used for flutter suppression control design. The DLR, UBRISTOL and SZTAKI applied different approaches for the control design. The DLR and SZTAKI controllers were assessed in the HIL environment. After the description and delay measurement of the HIL setup the possible mission scenarios and the implementation issues of the controllers were described. Finally, the hybrid SIL-HIL evaluation of DLR and SZTAKI flutter controllers was done. While the former was able to mitigate flutter if the system delay was close to its design delay even until the maximum achievable airspeed the latter was not able to stabilize the system even with lower time delay. So while the former is ready for HIL and possibly real flight testing the latter should be redesigned and improve to satisfy all criteria.

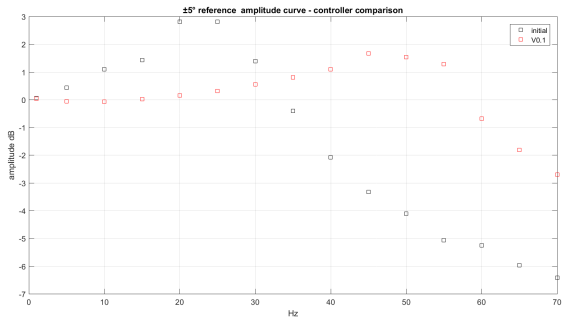

Fig. 39 Amplitude characteristics of different low level control loops on the custom direct-drive application

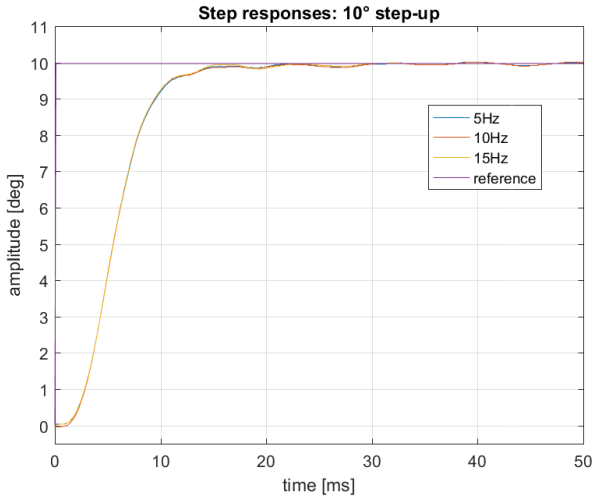

Fig. 40 Step response of the custom direct-drive system

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no 636307.

References

[1] FLEXOP, "Flutter Free FLight Envelope eXpansion for ecOnomical Performance improvement (FLEXOP)," Project of the European Union, Project ID: 636307, 2015-2018.

[2] PAAW, "Performance Adaptive Aeroelastic Wing Program," Supported by NASA NRA "Lightweight Adaptive Aeroelastic Wing for Enhanced Perfromace Across the Flight Envelope", 2014-2019.

[3] Zhao, W., Gupta, A., Miglani, J., Regan, C. D., Kapania, R. K., and Seiler, P. J., "Finite Element Model Updating of Composite Flying-wing Aircraft using Global/Local Optimization," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. https://doi.org/10.2514/6.2019-1814, URL https://doi.org/10.2514/6.2019-1814.

[4] Schmidt, D. K., Danowsky, B. P., Seiler, P. J., and Kapania, R. K., "Flight-Dynamics and Flutter Analysis and Control of an MDAO-Designed Flying-Wing Research Drone," *AIAA Scitech 2019 Forum*, AIAA, 2019.

[5] Kotikalpudi, A., Danowsky, B. P., Schmidt, D. K., Regan, C. D., and Gupta, A., "Real-Time Shape Estimation for a Small Flexible Flying-Wing Aircraft," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. https://doi.org/10.2514/6.2019-1818, URL https://doi.org/10.2514/6.2019-1818.

[6] Theis, J., Pfifer, H., and Seiler, P., "Robust Control in Flight: Active Flutter Suppression," *AIAA Science and Technology Forum*, 2016.

[7] Luspay, T., Ossmann, D., Wuestenhagen, M., Teubl, D., Baár, T., Pusch, M., Kier, T. M., Waitman, S., Ianelli, A., Marcos, A., Vanek, B., and Lowenberg, M. H., "Flight control design for a highly flexible flutter demonstrator," *AIAA Scitech 2019 Forum*, AIAA, 2019.

[8] Shamma, J. S., "Analysis and Design of Gain Scheduled Control Systems," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, 1988.

[9] Becker, G., "Quadratic Stability and Performance of Linear Parameter Dependent Systems," Ph.D. thesis, University of California, Berkeley, 1993.

[10] Moreno, C., Gupta, A., Pfifer, H., Taylor, B., and Balas, G., "Structural Model Identification of a Small Flexible Aircraft," *American Control Conference*, 2014, pp. 4379–4384.

[11] Gupta, A., Moreno, C. P., Pfifer, H., Taylor, B., and Balas, G. J., "Updating a finite element based structural model of a small flexible aircraft," American Institute of Aeronautics and Astronautics, 2015. https://doi.org/10.2514/6.2015-0903.

[12] Kotikalpudi, A., "Robust Flutter Analysis for Aeroservoelastic Systems," Ph.D. thesis, University of Minnesota, Twin Cities, 2017.

[13] Schmidt, D. K., "Stability Augmentation and Active Flutter Suppression of a Flexible Flying-Wing Drone," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2016, pp. 409–422. https://doi.org/10.2514/1.g001484.

[14] Meddaikar, Y. M., Dillinger, J., Klimmek, T., Krueger, W., Wuestenhagen, M., Kier, T. M., Hermanutz, A., Hornung, M., Rozov, V., Breitsamter, C., Alderman, J., Takarics, B., and Vanek, B., "Aircraft Aeroservoelastic Modelling of the FLEXOP Unmanned Flying Demonstrator," *AIAA Scitech 2019 Forum*, AIAA, 2019.

[15] Wuestenhagen, M., Kier, T., Meddaikar, Y. M., Pusch, M., Ossmann, D., and Hermanutz, A., "Aeroservoelastic Modeling and Analysis of a Highly Flexible Flutter Demonstrator," *2018 Atmospheric Flight Mechanics Conference*, AIAA, 2018.

[16] Takarics, B., Vanek, B., Kotikalpudi, A., and Seiler, P., "Flight control oriented bottom-up nonlinear modeling of aeroelastic vehicles," *2018 IEEE Aerospace Conference*, IEEE, 2018.

[17] Schmidt, D. K., *Modern Flight Dynamics*, McGraw-Hill, 2012. ISBN 9780073398112.

[18] Sodja, J., and et al., "Ground Testing of the FLEXOP Demonstrator Aircraft," *AIAA Scitech 2020 Forum (Submitted)*, American Institute of Aeronautics and Astronautics, Orlando, Florida, 2020.

[19] Luspay, T., Péni, T., Gőzse, I., Szabó, Z., and Vanek, B., "Model reduction for LPV systems based on approximate modal decomposition," *International Journal for Numerical Methods in Engineering*, Vol. 113, No. 6, 2017, pp. 891–909.

[20] Antoulas, A., *Approximation of Large-Scale Dynamical Systems*, Advances in Design and Control, SIAM, 2005.

[21] Adegas, F., Sonderby, I., Hansen, M. H., and Stoustrup, J., "Reduced-order LPV model of flexible wind turbines from high fidelity aeroelastic codes," *IEEE Conference on Control Applications*, 2013, pp. 424–429.

[22] Poussot-Vassal, C., and Roos, C., "Generation of a reduced-order LPV/LFT model from a set of large-scale MIMO LTI flexible aircraft models," *Control Engineering Practice*, Vol. 20, 2012, pp. 919–930.

[23] Burkard, R., Dell'Amico, M., and Martello, S., *Assignment Problems*, SIAM, 2009.

[24] Lovasz, L., and Plummer, M. D., *Matching Theory*, Annals of Discrete Mathematics, Vol. 29, North-Holland Mathematics Studies, 1986.

[25] Manning, C. D., and abd H. Schütze, P. R., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[26] Wood, G. D., "Control of Parameter-Dependent Mechanical Systems," Ph.D. thesis, University of Cambridge, 1995.

[27] Vinnicombe, G., "Measuring Robustness of Feedback Systems," Ph.D. thesis, Department of Engineering, University of Cambridge, 1993.

[28] Takarics, B., and Vanek, B., "Tensor Product Model-based Robust Flutter Control Design for the FLEXOP Aircraft," *IFAC-PapersOnLine*, Vol. 52, No. 12, 2019, pp. 134–139. https://doi.org/10.1016/j.ifacol.2019.11.149, URL https://doi.org/10.1016/j.ifacol.2019.11.149.

[29] Tanaka, K., and Wang, H. O., *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, John Wiley & Sons, Inc., 2001.

[30] Pusch, M., Ossmann, D., and Luspay, T., "Structured Control Design for a Highly Flexible Flutter Demonstrator," *Aerospace*, Vol. 6, No. 3, 2019, p. 27.

[31] Pusch, M., "Aeroelastic Mode Control using H_2-optimal Blends for Inputs and Outputs," *Proc. of Guidance, Navigation, and Control Conference, AIAA SciTech Forum*, AIAA, Kissimmee, FL, 2018. https://doi.org/10.2514/6.2018-0618.

[32] Apkarian, P., Dao, M.-N., and Noll, D., "Parametric Robust Structured Control Design," *IEEE Transactions on Automatic Control*, Vol. 60, No. 7, 2015, pp. 1857–1869. https://doi.org/10.1109/TAC.2015.2396644.

[33] Waitman, S., and Marcos, A., "Active flutter suppression: non-structured and structured H∞ design," *IFAC-PapersOnLine*, Vol. 52, No. 12, 2019, pp. 146–151. https://doi.org/10.1016/j.ifacol.2019.11.184, URL https://doi.org/10.1016/j.ifacol.2019.11.184.

[34] Waitman, S., and Marcos, A., "H infinity control design for active flutter suppression of flexible-wing UAV demonstrator," *Journal of Guidance, Control, and Dynamics*, Vol. accepted for publication, 2019.

[35] Apkarian, P., and Noll, D., "Nonsmooth H∞Synthesis," *IEEE Transactions on Automatic Control*, Vol. 51, No. 1, 2006, pp. 71–86. https://doi.org/10.1109/tac.2005.860290, URL https://doi.org/10.1109/tac.2005.860290.

[36] Marcos, A., and Balas, G. J., "A robust integrated controller/diagnosis aircraft application," *International Journal of Robust and Nonlinear Control*, Vol. 15, No. 12, 2005, pp. 531–551. https://doi.org/10.1002/rnc.1010, URL https://doi.org/10.1002/rnc.1010.

[37] Simplício, P., Marcos, A., Joffre, E., Zamaro, M., and Silva, N., "Synthesis and analysis of robust control compensators for Space descent & landing," *International Journal of Robust and Nonlinear Control*, Vol. 28, No. 13, 2018, pp. 3871–3892. https://doi.org/10.1002/rnc.4109, URL https://doi.org/10.1002/rnc.4109.

[38] Navarro-Tapia, D., Marcos, A., Simplício, P., Bennani, S., and Roux, C., "Legacy recovery and robust augmentation structured design for the VEGA launcher," *International Journal of Robust and Nonlinear Control*, Vol. 29, No. 11, 2019, pp. 3363–3388. https://doi.org/10.1002/rnc.4557, URL https://doi.org/10.1002/rnc.4557.

[39] Marcos, A., and Sato, M., "Flight testing of an structured H-infinity controller: An EU-Japan collaborative experience," *2017 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2017. https://doi.org/10.1109/ccta.2017.8062768, URL https://doi.org/10.1109/ccta.2017.8062768.

[40] Patartics, B., Vanek, B., and Seiler, P., "Structured Robust Synthesis with Parameter-Dependent D-Scales," *American Control Conference*, 2019, pp. 1800–1805.

[41] Patartics, B., Seiler, P., and Vanek, B., "GitHub repository of the MATLAB implementation of the structured design algorithm," , 2019. URL https://github.com/Patartics-Balint/wcgmin.

[42] Pataritcs, B., Lipták, G., Luspay, T., Seiler, P., Takarics, B., and Vanek, B., "Application of structured robust synthesis for flexible aircraft flutter suppression," *IEEE Transactions on Control Systems Technology*, 2019 (submitted).

[43] Stein, G., "Respect the unstable," *IEEE Control Systems Magazine*, Vol. 23, No. 4, 2003, pp. 12–25.

[44] Pusch, M., Ossmann, D., and Luspay, T., "Structured Control Design for a Highly Flexible Flutter Demonstrator," *Aerospace (MDPI)*, Vol. 6, No. 27, 2019.

[45] Patartics, B., Luspay, T., Péni, T., Takarics, B., Vanek, B., and Kier, T., "Parameter varying flutter suppression control for the BAH jet transport wing," *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 8163 – 8168. https://doi.org/https://doi.org/10.1016/j.ifacol.2017.08.1263, URL http://www.sciencedirect.com/science/article/pii/S2405896317317780, 20th IFAC World Congress.

[46] Patartics, B., Peni, T., and Vanek, B., "Flexibilis szárnyú repülőgép aktív flatter szabályozása," *Repüléstudományi közlemények*, Vol. 30, No. 1, 2018.

[47] Sodja, J., Breuker, R. D., Meddaikar, Y. M., Dillinger, J., Soal, K., Govers, Y., Krueger, W. R., Georgopoulos, P., Koimtzoglou, C., Roessler, C., Koeberle, S., Bartasevicius, J., Dániel Teubl4, L. G., Tóth, S., Nagy, M., Balogh, D., Jásdi, M., and andBálint Vanek, P. B., "Ground Testing of the FLEXOP Demonstrator Aircraft," *In proc. of AIAA Scitech Forum 2020*, 2020.

A. Input and wing flutter energies

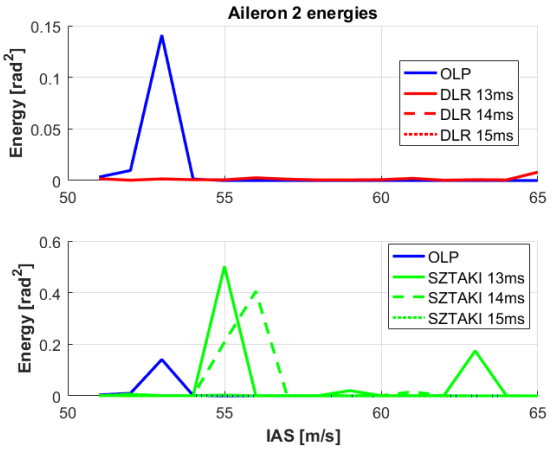

Fig. 41 Energy content of baseline aileron actuator (Nr. 2)

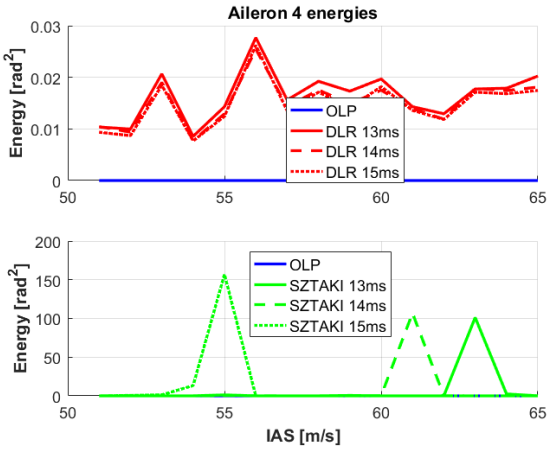

Fig. 42 Energy content of flutter aileron actuator (Nr. 4)

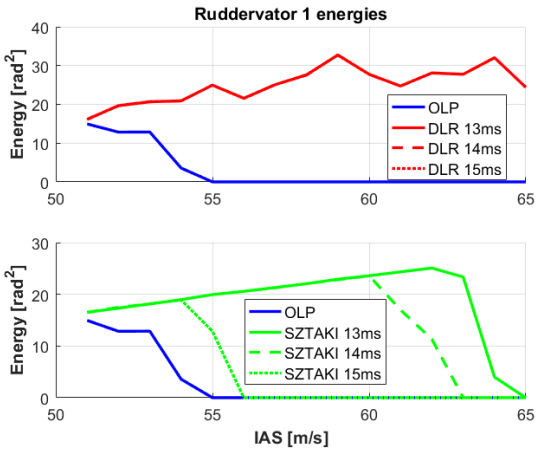

Fig. 43 Energy content of baseline ruddervator actuator (Nr. 1)

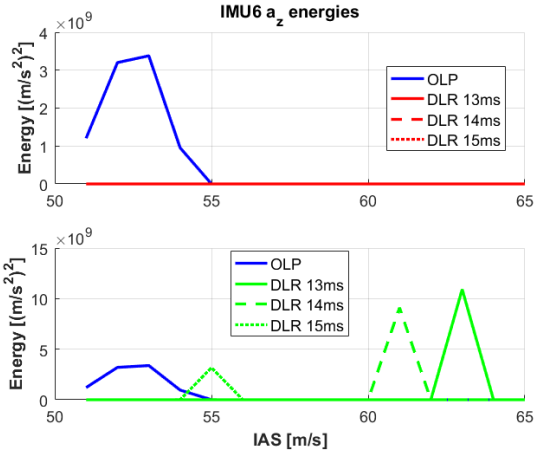

Fig. 44 Energy content of wing IMU (Nr. 6) acceleration

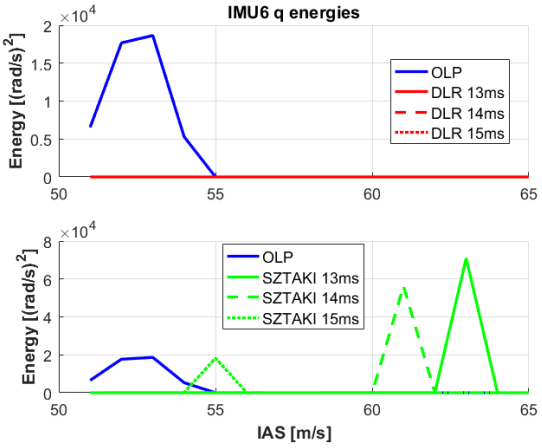

Fig. 45 Energy content of wing IMU (Nr. 6) pitch rate

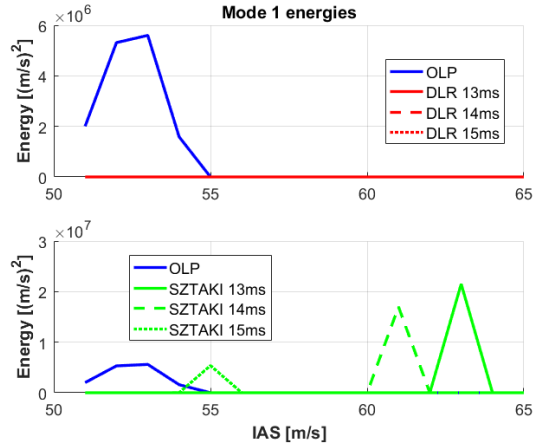

Fig. 46 Energy content of wing flexible mode 1

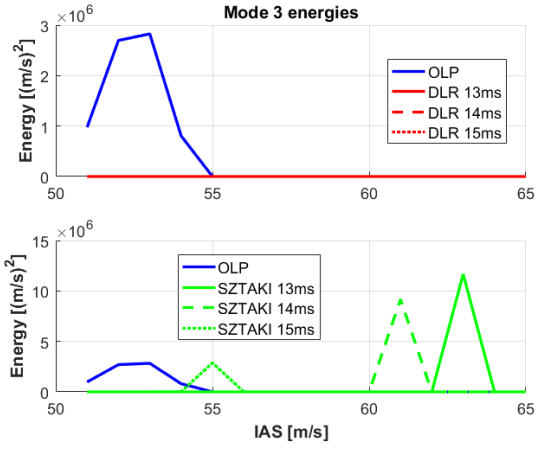

Fig. 47 Energy content of wing flexible mode 3

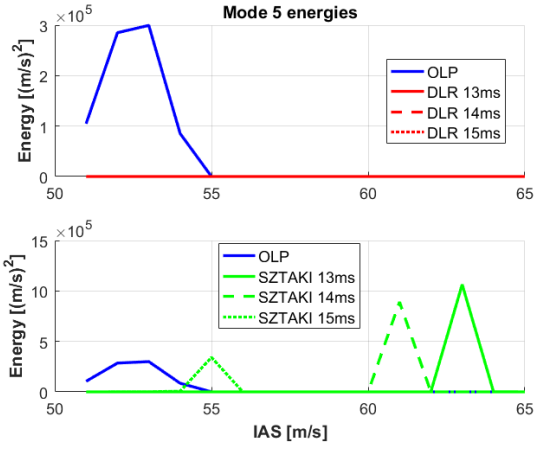

Fig. 48 Energy content of wing flexible mode 5