International Forum on Aeroelasticity and Structural Dynamics
IFASD 2024
17-21 June 2024, The Hague, The Netherlands

FLEXIBLE AIRCRAFT CONCEPTUAL CO-DESIGN BASED ON THE RCE FRAMEWORK

Béla Takarics[1], Bálint Patartics[1], Bálint Vanek[1], Fanglin Yu[2], Yasser M. Meddaikar[3], Matthias Wuestenhagen[4], Thiemo Kier[4]

[1]HUN-REN Institute for Computer Science and Control
Kende utca 13-17, 1111 Budapest, Hungary
takarics@sztaki.hu
patartics@sztaki.hu
vanek@sztaki.hu

[2]Technical University of Munich
Boltzmannstrasse 15, 85748 Garching, Germany
fanglin.yu@tum.de

[3]DLR - Institute of Aeroelasticity
Bunsenstrasse 10, 37073 Goettingen, Germany
Muhammad.Meddaikar@dlr.de

[4]DLR - Institute of System Dynamics and Control
Münchener Strasse 20, 82234 Wessling, Germany
Matthias.Wuestenhagen@dlr.de
Thiemo.Kier@dlr.de

Keywords: reduced order modelling, conceptual aircraft design, active control

Abstract: In a traditional aircraft design process, the air-frame is designed first, followed by synthesizing the control system. This sequential method does not guarantee the best possible closed-loop performance. The goal of the paper is to propose automatic control design methods for flexible aircraft. Such approach enables the inclusion of the control design algorithms into the multidisciplinary design optimization (MDO) of aircraft design. In such an extended MDO framework, called co-design, the sizing, structural dynamics, aerodynamics and the controllers of the aircraft are optimized in one single step. Co-design allows the usage of control technologies early in the conceptual preliminary design stage of aircraft design. A demonstrator aircraft T-Flex serves as a test bed for the conceptual co-design. DLR's Remote Component Environment (RCE) framework is used as the backbone for the MDO toolchain implementation. The control design algorithms of the co-design considered in the paper are the baseline and the flutter suppression controllers, which require a control oriented aeroservoelastic model. The modeling is done via the bottom-up modeling approach and the resulting control oriented models are given in the linear parameter-varying (LPV) framework. The baseline control system features a classical cascade flight control structure with scheduled control loops to augment the lateral and longitudinal axis of the aircraft. The control loops use scheduled elements of proportional-integral-derivative (PID) controller structures. The flutter controller aims to mitigate the undamped oscillations of the wings that occur if the aircraft is flying beyond the flutter speed. Parametric and dynamic uncertainties are accounted for in the control design. The objective of the design is to minimize the sensitivity function of the closed-loop while limiting

the bandwidth of the controller to prevent the excitation of high-frequency dynamics. The effect of the following parameters on the conceptual co-design is investigated: wing sweep angle between 0 and 30 degrees and flutter mass between 0 and 0.4 kg. It is examined how these parameters affect the flutter modes of the aircraft and the performances of the baseline and flutter suppression controllers.

1 INTRODUCTION

In order to reduce harmful emissions and operating costs, one of the main directions of future aircraft design is to increase the wing aspect ratio and decrease weight. Such air-frames however, have increased aeroelastic behavior that are often prone to instability. Control design for such flexible air vehicles, which might require active flutter suppression controllers, is a greater challenge than for classical rigid-like aircraft. There exist several recent research projects investigating control design methodologies for flexible aircraft. These are the Performance Adaptive Aeroelastic Wing (PAAW) project in the USA, [1] and the Flutter Free FLight Envelope eXpansion for ecOnomical Performance improvement (FLEXOP) and Flight Phase Adaptive Aero-Servo-Elastic Aircraft Design Methods (FLIPASED) projects in the EU, [2, 3]. A conceptual aircraft design is commonly obtained via a classical multidisciplinary design optimization (MDO) process, where the structure and aerodynamics are optimized in one common step. The control design comes at a later stage. In the case of flexible aircraft, such a sequential structure-aero and control design optimization might lead to a sub-optimal performance aircraft because the increased airframe flexibility leads to increased complexity in the control design step. The challenges related to future conceptual aircraft designs this paper is focusing on are therefore twofold. First, more advanced control loops are expected to be addressed and second, the classical MDO approach might not be sufficient for airframe optimization. It is important to include the control design aspects early-on in the preliminary stage of conceptual aircraft design for air vehicles with increased structural flexibility. This can be achieved by including the control design into the MDO process of aircraft design. Such approach, where the aerodynamics, structural dynamics and the control design are optimized in a common step is called co-design [4–7]. In case of co-design for flexible aircraft, on top of the baseline controller design, the active flutter suppression control design needs to be investigated as well.

One of the main challenges of co-design is finding a suitable control oriented modelling and control design structure that can be included into the MDO process. The linear parameter-varying (LPV) [13, 14] framework is well suited for control design for flexible aircraft, for which a suitable low order control oriented LPV aeroservoelastic (ASE) model for flexible aircraft is crucial. The ASE model of an aircraft is typically constructed based on the integration of aerodynamics, structural dynamics and flight dynamics subsystems, [8–11]. The models of the subsystems are developed separately which are then combined. This way the ASE model is formed. One of the challenges with such ASE models is that they are typically of too high order for control design. Therefore, low order control oriented model is required. The main approach for the control oriented model development is the ”bottom-up” modeling [12]. The key idea of the bottom-up modeling is the following. The subsystems of the ASE model in general have simpler structure than the nonlinear ASE model. Therefore, the subsystems containing the structural dynamics and aerodynamics model can be reduced by simpler, more tractable reduction techniques. Creating the ASE model from the reduced subsystems leads to the control oriented model which is suitable for control design. If such modelling framework is to be used for co-design, the control oriented model generation needs to be done in an automatic fashion. The control design step for flexible aircraft typically involves a lot of manual tuning and trial and error steps until the controller with the highest performance and robustness is achieved. In addition, the control synthesis often has a high computational load. In order to include the control design step into the MDO, it needs to have small computational load and the algorithm needs to be robust against the aerodynamic and structural dynamic changes that occur during the optimization process. The whole MDO process needs to operate in an automatic fashion, without hand tuning. In addition to the control synthesis, performance and robustness analysis of the resulting controllers needs to be carried out in an automated fashion as well. Some of the automation aspects of the model generation and control design are elaborated in [23].

The main goal of the paper is to develop an extended MDO toolchain that is capable of co-design for flexible aircraft. By this approach the airframe parameters and the controllers can be optimized in a common step. The aircraft under consideration is the T-Flex aircraft, which was designed and built in the FLEXOP and FLIPASED projects. This aircraft serves as a test bed for active flutter suppression control methods because it has two unstable flutter modes within the flight envelope. Therefore, it serves as a good candidate for the co-design toolchain. The T-Flex demonstrator UAV is shown in Figure 1 and it is presented in Section 2. The co-design toolchain developed in this paper, described in Section 2, is implemented and tested in the Remote Component Environment (RCE) environment [15], Section 3. The MDO co-design toolchain is utilized to assess the effects of several aircraft design parameters on the flutter modes and the control design related consequences of these parameters. For this, first the control oriented model development, control design and analysis algorithms are automated and implemented into the MDO toolchain, Sections 4 and 5. Second, a parameter study is executed and the effects are evaluated, Section 6. The parameter study considers the variations in the flutter mass and wing sweep angles.

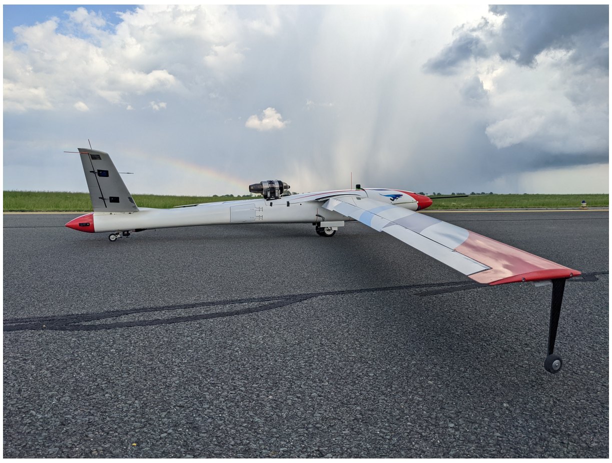


Figure 1: T-Flex demonstrator aircraft

2 REFERENCE AIRCRAFT AND MDO TOOLCHAIN ARCHITECTURE

The reference aircraft of the MDO toolchain is the T-Flex demonstrator of the FLEXOP and FLIPASED projects [2,3]. This aircraft has the following main characteristics. The aircraft has a wingspan of 7 m and aspect ratio of 20. The aircraft has a 300 N jet engine. The empennage is configured as a V-tail and each wing has 4 control surfaces, [16]. The outer control surfaces are used for flutter suppression, see Figure 2.

The reference aircraft has two unstable aeroelastic modes. The first aeroelastic mode (symmetric) goes unstable at 52 m/s and 50.2 rad/s and the second (asymmetric) at 55 m/s and 45.8 rad/s. In order to have sufficient bandwidth, custom made actuators are designed for the aircraft. In

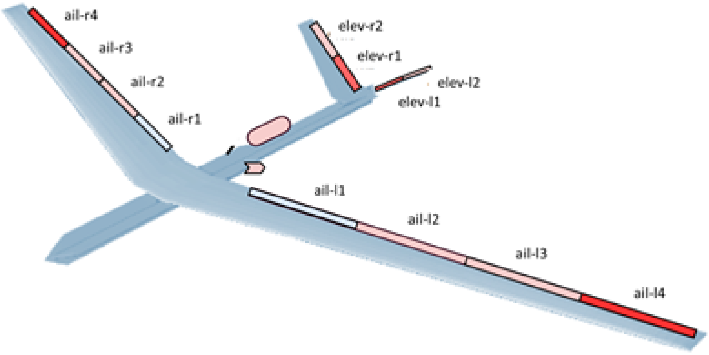


Figure 2: T-Flex aircraft control surface configuration

addition to the GPS and air data probe, the aircraft has inertial measurement units (IMUs) at the center of gravity and in the wings as shown in Figure 3.

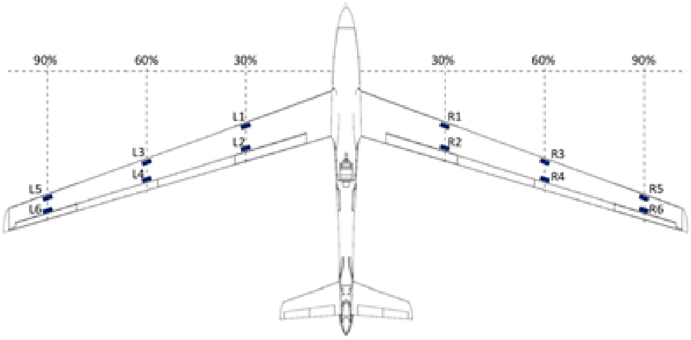


Figure 3: T-Flex aircraft sensor configuration

The central component in this scenario is the MDO toolchain, which encompasses its own optimization process and returns to the Common Parametric Aircraft Configuration Schema (CPACS) generation block after each iteration. The overall architecture of the MDO toolchain is illustrated in Figure 4. The co-design architecture starts with the design variables as the input.

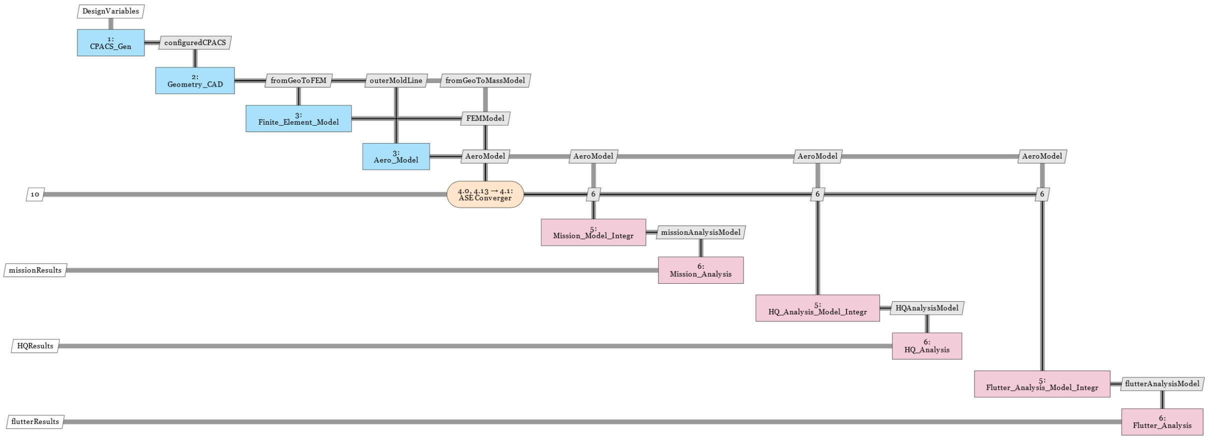


Figure 4: MDO Toolchain for demonstrator T-FLEX

These in the present case are the wing sweep angle Λ and the flutter mass m_{fl} with nominal values $\Lambda_{nom} = 20°$ and $m_{fl_{nom}} = 0.24/, kg$, respectively. The CPACS description file is generated based on the design parameters in Block 1 and the aircraft geometry is updated in Block 2. Blocks 3 then generates the finite element model (FEM) and the aerodynamic model. The

ASE model is then generated in Block 4, based on which the Mission Model, the Handling Quality (HQ) Model and the Flutter models are constructed in Block 5. The baseline (HQ) and flutter suppression controllers are designed and evaluated in Block 6. The primary objective of the MDO toolchain is to demonstrate the enhancements achieved through optimization, encompassing aircraft geometry, sizing, modeling, and control design concurrently, in comparison to the reference aircraft. The relative increase of the flutter speed with active flutter suppression control is chosen as the main cost function of the mission analysis. The subsequent sections will provide a concise overview of the function blocks within the MDO toolchain and the standard tools employed.

3 REMOTE COMPONENT ENVIRONMENT

DLR's Remote Component Environment (RCE) [17, 18] is an open-source software environment for defining and executing workflows containing distributed simulation tools by integrating them into a peer-to-peer network. The RCE framework is used as the backbone for the MDO toolchain implementation. The following description has been taken from the related publication by the main developers, Boden et al. [17, 18]. RCE is being developed primarily by DLR and has been used in various engineering projects, including several aerospace projects dealing with MDO and multidisciplinary analysis (MDA). RCE has several advantages that can help to achieve more reusable multidisciplinary processes. The workflow is composed of built-in and user-defined components. Disciplinary tools are integrated as standalone components, with defined inputs and outputs, and then distributed over the network. While executing the workflow, data dependencies between the components are automatically detected, and a component is executed as soon as all its input data is available. Thus, multiple components can run at the same time. The components of a multidisciplinary process can also be executed in a distributed manner, where the tools are located on different machines with possibly different operating systems. Once configured, the peer-to-peer network is automatically established between the RCE instances running on different machines, making components visible and executable even between instances that are only connected indirectly. The distributed execution capability alleviates tool deployment issues, Figure 5, including those related to the protection of intellectual property.

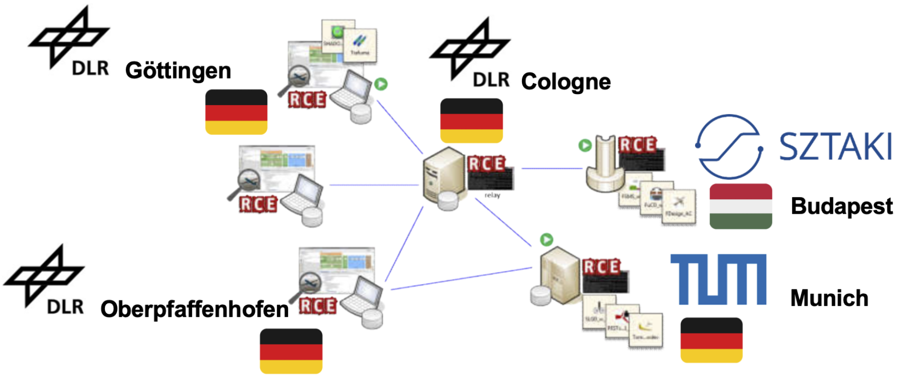


Figure 5: Distributed RCE workflow

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

The final implementation of the workflow can be seen in Figure 6. The figure also indicates the responsible partners within the FLIPASED project. Technical University of Munich (TUM) handles the design parameter inputs, the geometry, FEM model and the aerodynamic panel generation (Blocks 1, 2 and 3 of Figure 4., blue blocks in Figure 6.). DLR is responsible for the ASE model generation and for the open loop mission analysis (Block 4 in Figure 4., orange and green blocks in Figure 6.), while the Institute for Computer Science and Control (SZTAKI) is responsible for the control oriented modelling, control synthesis and analysis blocks (Blocks 5 and 6 in Figure 4., pink blocks in Figure 6.).

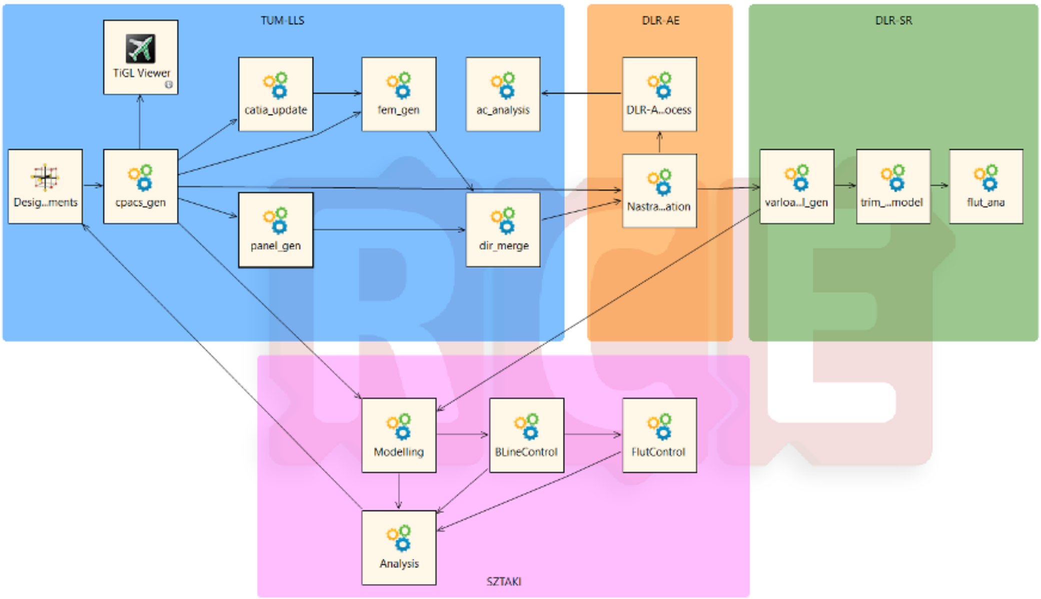


Figure 6: Implemented RCE workflow

4 AIRCRAFT MODELLING

4.1 Dynamical high complexity model construction block (TUM)

The TUM block takes the design parameters as input and constructs a high fidelity nonlinear model of the aircraft. From this high fidelity model the control oriented models for the various control design applications can be derived. Catia model of wing -1 is reconstructed with parametrized platform and structural layout. The current CAD model is fully capable of handling the design parameters. The geometrical modelling process is automated with the Catia macro language. Structure of wing -1 was modelled in HyperMesh (Figure 7.). All the model generation operation was programmed with HyperMesh native macro language TCL. Currently the geometrical and structural modelling tools are integrated in the RCE framework and works automatically without human intervention.

The aerodynamic modelling tool for aeroelastic analysis is developed with PyNastran (Figure 8). It is implemented with a CPACS interface to ease the data input and has a default Nastran output. The tool is also integrated into RCE. The block provides the FEM and the aerodynamic panels as output.

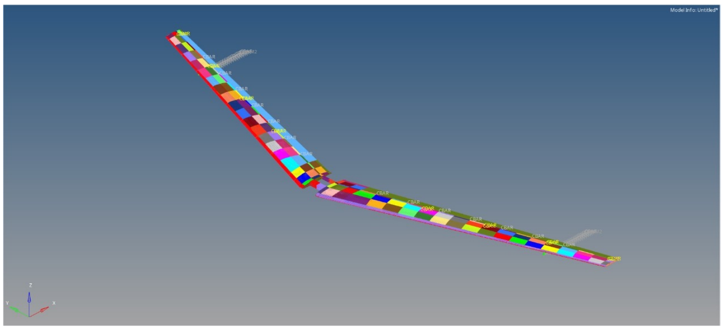


Figure 7: Wing FEM generation

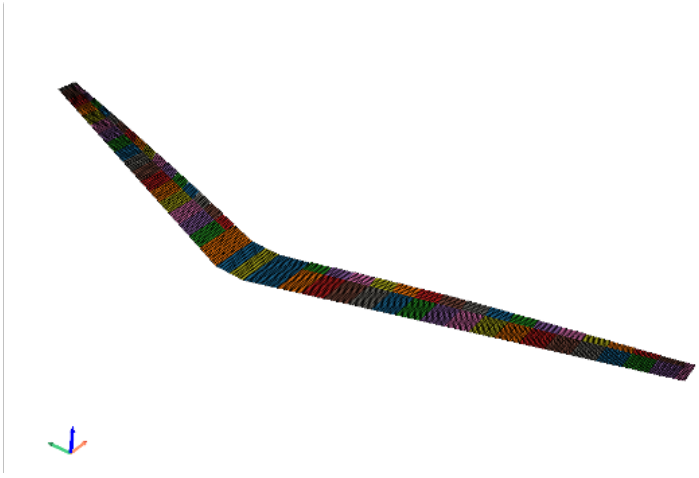


Figure 8: Wing DLM generation

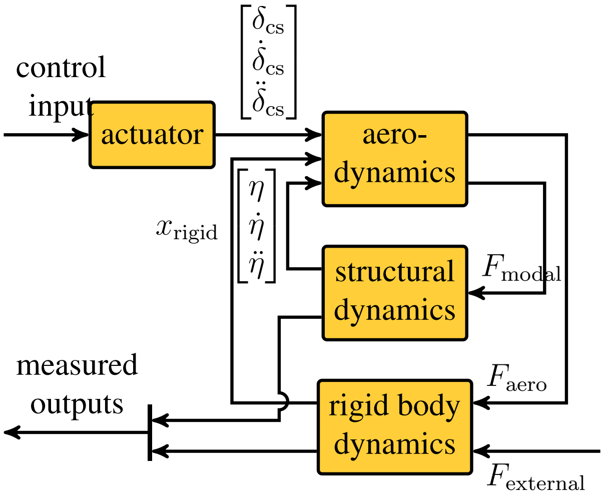
4.2 Nonlinear model generation and open loop flutter analysis (DLR)

The outputs of the TUM block serve as the input for the DLR blocks. The FEM and aerodynamics panels are integrated into Nastran and a Simulink nonlinear model is generated. Details of the aeroservoelastci modelling can be found in [11, 19]. Furthermore, different methods of flutter analysis are applied for the open-loop model, in order to validate the used tools. The Simulink model representing the nonlinear flexible aircraft dynamics can be used in order to determine the open-loop flutter speeds and frequencies. The aircraft is trimmed and linearised for a couple of flight conditions. Especially, differences in flight speed are of interest for the demonstrator aircraft, as they have the biggest effect on the aeroelastic modes. The set of linearised state-space models is then analysed with respect to their eigenvalues. As soon as a pole crosses the imaginary axis and migrates to the right half plane, unstable flutter becomes an issue. Based on the trimmed airspeed of the linearised systems it can then be determined what the flutter speed is.

4.3 Control oriented modelling of the T-Flex demonstrator aircraft (SZTAKI)

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

Flexible aircraft are typically modelled using subsystems. The structural dynamics model, the



control input
actuator
$\begin{bmatrix}\delta_{\mathrm{cs}}\\ \dot{\delta}_{\mathrm{cs}}\\ \ddot{\delta}_{\mathrm{cs}}\end{bmatrix}$
aero-dynamics
x_{rigid}
$\begin{bmatrix}\eta\\ \dot{\eta}\\ \ddot{\eta}\end{bmatrix}$
structural dynamics
F_{modal}
measured outputs
F_{aero}
rigid body dynamics
F_{external}

Figure 9: ASE subsystem interconnection

aerodynamics model and the flight mechanics model are combined to form the aeroservoelastic (ASE) model. Such subsystem interconnection is depicted in Figure 9. These ASE models in general are of too high order for control design, therefore, model order reduction is required. One approach applied for the MDO process is the bottom-up modeling approach, [11, 12, 19].

The key idea of the bottom-up modeling is the following. The subsystems of the ASE model in general have simpler structure than the nonlinear ASE model. Therefore, the subsystems containing the structural dynamics and aerodynamics model can be reduced by simpler, more tractable reduction techniques. Combining these reduced order subsystems results in a low order nonlinear ASE model upon which a nominal, low order, control oriented models can be obtained. The control oriented models are based on the LPV framework, [13, 14]. The LPV framework can serve as a good approach to model ASE systems for control design. The benefits of utilizing the LPV framework are the following; it can capture the parameter varying dynamics of the aircraft and many of the linear time-invariant (LTI) control design techniques have been extended to LPV systems. An LPV system is described by the state-space model [13, 20]

$$\dot{x}(t) = A(\rho(t))\ x(t) + B(\rho(t))\ u(t) \tag{1a}$$
$$y(t) = C(\rho(t))\ x(t) + D(\rho(t))\ u(t) \tag{1b}$$

with the continuous matrix functions $A\colon \mathcal{P} \to \mathbb{R}^{n_x \times n_x}$, $B\colon \mathcal{P} \to \mathbb{R}^{n_x \times n_u}$, $C\colon \mathcal{P} \to \mathbb{R}^{n_y \times n_x}$, $D\colon \mathcal{P} \to \mathbb{R}^{n_y \times n_u}$, the state $x\colon \mathbb{R} \to \mathbb{R}^{n_x}$, output $y\colon \mathbb{R} \to \mathbb{R}^{n_y}$ input $u\colon \mathbb{R} \to \mathbb{R}^{n_u}$, and a time-varying scheduling signal $\rho\colon \mathbb{R} \to \mathcal{P}$, where $\mathcal{P}$ is a compact subset of $\mathbb{R}^N$. The system is called quasi LPV model if the parameter vector ρ includes elements of the state vector x. The system matrix $S(\rho(t))$ is defined as

$$S(\rho(t)) = \begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} \tag{2}$$

In a grid-based LPV representation ([20]), the system is described as a collection of LTI models $(A_k, B_k, C_k, D_k) = (A(\rho_k), B(\rho_k), C(\rho_k), D(\rho_k))$ obtained from evaluating the LPV model at a finite number of parameter values $\{\rho_k\}_1^{n_{\text{grid}}} = \mathcal{P}_{\text{grid}} \subset \mathcal{P}$. Using the bottom-up modeling approach, a full order and a low order LPV model is obtained. The full order model is constructed without reducing the subsystems, while the low order model is obtained by reducing the structural dynamics and aerodynamics subsystems. The main measure of the accuracy of the low order model is the ν-gap metric, [21]. The ν-gap between the full and low order models is calculated frequency wise at each grid point.

The modeling block in RCE takes the structural dynamics (M_{hh}, K_{hh}, B_{hh}) and aerodynamics data (Q_{hh}) as input via CPACS from DLR. These parameters are expected to change due to the MDO optimization, the rest of the model properties are assumed to be fixed.

4.3.1 Reduction of the structural dynamics model

The structural dynamics of the aircraft are of the form

$$\mathcal{M}\ddot{\eta} + \mathcal{C}\dot{\eta} + \mathcal{K}\eta = F_{\text{modal}} \tag{3}$$

where F_{modal} is the force acting on the structure in modal coordinates, $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{K}$ are the modal mass, damping and stiffness matrices respectively. The full order structural model contains 50 modes, thus the structural dynamics model consists of 100 states. The structural dynamics model is an LTI system, thus state truncation can be applied.

4.3.2 Reduction of the aerodynamics model

The aerodynamic lag terms take the state-space form

$$\begin{aligned} \dot{x}_{\text{aero}} &= \frac{2V_{\text{TAS}}}{\bar{c}} A_{\text{lag}} x_{\text{aero}} + B_{\text{lag}} \begin{bmatrix} \dot{x}_{\text{rigid}} \\ \dot{\eta} \\ \dot{\delta}_{\text{cs}} \end{bmatrix} \\ y_{\text{aero}} &= C_{\text{lag}} x_{\text{aero}} \end{aligned} \tag{4}$$

where V_{TAS} is the true airspeed, x_{rigid} is the rigid body state, η is the modal state of the structural dynamics, δ_{cs} is the control surface deflection and $\bar{c}$ is the reference chord. The full order model consists of 1040 lag states. Using the aerodynamics model given by A_{lag}, B_{lag} and C_{lag} in (4) an LTI balancing transformation matrix T_{b} is computed. The balanced states of the aerodynamic model with the smallest Hankel singular values are residualized, leading to a reduced order aerodynamics model.

The initial model order reduction produced the following results. The structural dynamics model can be reduced in the following way. In order to keep the ν-gap between the high fidelity and the low order model low the first six structural modes are retained. Removing further structural modes results in a large increase in the ν-gap. This way, a 12 state structural dynamics model can be obtained from the 100th order model. In case of the aerodynamics model, retaining two lag states results in a low order model with acceptable accuracy. The resulting nonlinear ASE bottom-up model has 26 states that consists of 12 rigid body states, 12 structural dynamics states, 2 aerodynamic lag states. Note, that the actuator dynamics are not included in the control oriented model. The ν-gap between the full order and the reduced order model of the full order aircraft model for different airspeed values is given in Figure 10.

4.3.3 Uncertain low order model

The next step is to develop uncertain LPV models of the aircraft. Uncertain models can be developed by extending the structural dynamics model with the uncertain parameters. These uncertainties appear in the mass matrix $\mathcal{K}$ and in the damping matrix $\mathcal{C}$ in (3) of the nonlinear ASE model and are denoted by $\delta_{\mathcal{K}}$ and $\delta_{\mathcal{C}}$, respectively. Based on this uncertain, nonlinear

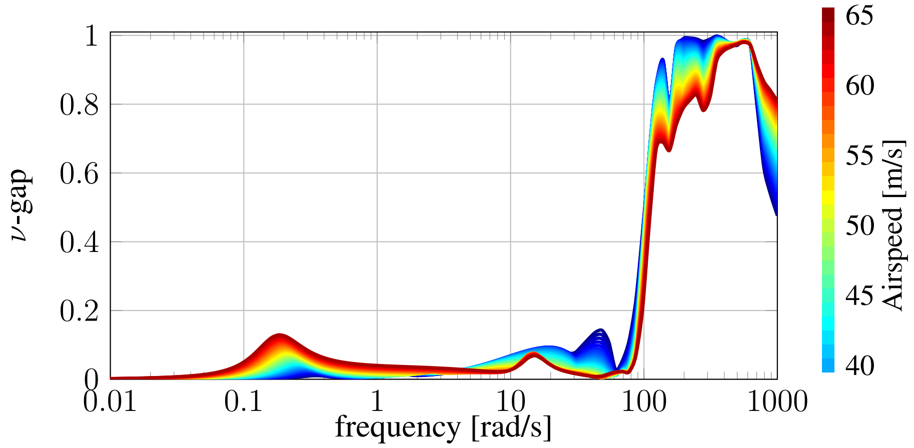


Figure 10: ν-gap values between the nominal low order and high-fidelity models

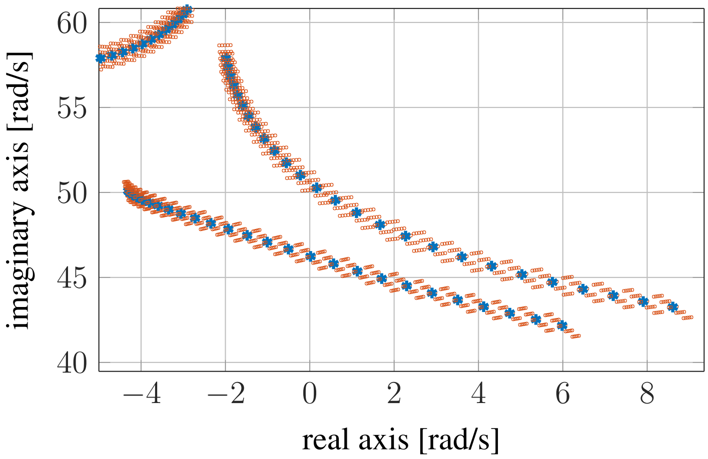


Figure 11: Uncertainty of the flutter modes: nominal model (blue), uncertain (red)

model a grid-based uncertain LPV model is constructed. The grid-based uncertain LPV model is obtained over a 3 dimensional grid. The grid consists of 31 equidistant points of the airspeed between 30m/s and 70m/s, 3 points of the natural frequency in the structural dynamics between $\pm 1\%$ of the nominal value, and 3 points of the damping in the structural dynamics between $\pm 10\%$ of the nominal value. This results in a total of $31 \times 3 \times 3 = 279$ grid points. The scheduling parameter ρ can then be defined as

$$\rho = \begin{bmatrix} \rho_{V_{TAS}} \\ \delta_{\mathcal{K}} \\ \delta_{\mathcal{C}} \end{bmatrix} \tag{5}$$

where $\rho_{V_{TAS}}$ is a measured parameter and $\delta_{\mathcal{K}}$ and $\delta_{\mathcal{C}}$ are unmeasured. These uncertainties have a significant effect on the flutter speeds and frequencies. The nominal and uncertain flutter modes of the control oriented LPV model are shown in Figure 11.

4.3.4 Modeling block resilience

As it can be seen, the bottom-up modeling approach involves a certain degree of heuristics. These heuristic steps include the selection of the structural dynamics states to retain and setting the the number of retained aerodynamic lag states. These parameters are hand tuned for the initial, reference aircraft model. The modeling tool needs to be adopted to the collaborative design in this respect. This means that the retained the initial structural modes to be retained are the ones of the reference aircraft. However, it is crucial that after every MDO iteration, the ν-gap metric is analyzed and that it does not exceed a threshold value. If this value is exceeded, it means that the bottom up-model is not accurate enough. Therefore, at the expense of increasing the order of the resulting model, additional structural modes need to be retained. The number of retained modes is increased until the ν-gap values are satisfactory. A similar approach is used for the order of the lag state aerodynamics model. In this case the number of the retained lag states is increased until a satisfactory ν-gap level is obtained.

The modeling block provides two LPV models, one for the baseline control design (RigACModel) and one for the flexible control design (FlexACModel). The FlexACModel is the low order, uncertain LPV model of the aircraft obtained by the steps described above. The RigACModel is obtained from the nominal low order aircraft model by rezidualizing the structural and lag state dynamics. This model serves for the baseline control design, containing only the 12 rigid body states. These resulting models are saved in the ToolSpecific section of CPACS.

5 CONTROL DESIGN AND ANALYSIS

5.1 Baseline control design functions (SZTAKI)

The starting point for the baseline control design is the baseline design for the flight tests. However, some modifications had to be carried out. The reason for this is that flight test controllers required a lot of hand tuning to achieve the highest performance. The key idea is to simplify these requirements in order to achieve controller design alrorithms that can run in an automatic fashion.

The baseline control design is based on the LPV model obtained in the model integration block, that has 12 rigid body state and the actuator dynamics. The baseline control design takes the actuator dynamics and the baseline control design model RigACModel as inputs via CPACS. The baseline control system features a classical cascade flight control structure with scheduled control loops to augment the lateral and longitudinal axis of the aircraft, see Figure 12. The control loops use scheduled elements of proportional-integral-derivative (PID) controller structures with additional roll-offs in the inner loops to ensure that no aeroelastic mode is excited by the baseline controller. Scheduling with indicated airspeed V_{ias} is used to ensure an adequate performance over the velocity range from 40 m/s to 70 m/s. Structurally the controller consists of several loops targeting different dynamical modes. Accordingly, intuitive design specification for the loops can be formulated by the user in terms of settling times, reference tracking or robustness margins. The control design itself automatically optimizes the corresponding gains, in order to satisfy the specified design goals. Once the optimization found a feasible solution it provides the corresponding control gains and control structure which is then used for the numerical analysis. The main algorithms of the baseline control design based on the approach described in [22]. The tool used to obtain the respective PID gains is `systune` of Matlab, which provides intuitive access to performance specifications and which well suited for implementation as automatic control design algorithm. In addition, several steps are included in order to make the automatic control design algorithms more robust against model variations resulting from the MDO process. These steps are detailed the controller description.

5.1.1 Inner lateral loop design

The inner lateral loop takes the LPV model containing the lateral states, inputs and outputs. Time and frequency domain control performance specifications are given. The time domain

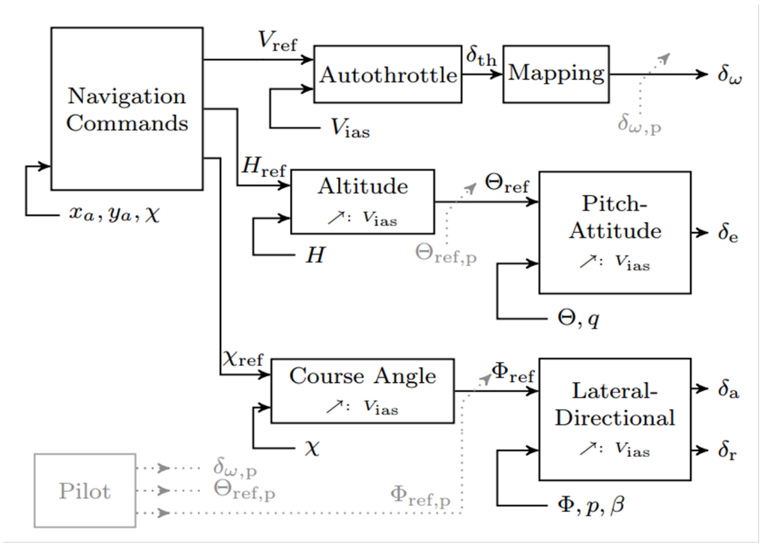


Figure 12: Baseline control architecture

performance requirements are formulated as a step reference system tracking. The reference system is given as a second order system

The ideal model is given as

$$G_{id_{lat}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $\omega_n = 0.7 \times \omega_{n_{roll}}$ is the natural frequency and $\zeta = 0.8$ is the damping factor of the ideal model. $\omega_{n_{roll}}$ is the roll frequency of the lateral model. Such approach improves the robustness of the control design approach with the expense of achieving a sub-optimal performance. The frequency domain specifications are $5dB$ gain and $40°$ phase margins. In case the performances cannot be met, the design is repeated with design criteria relaxed by 10% until a feasible controller is found.

5.1.2 Outer lateral loop design

Once a feasible inner controller is designed, the controller is connected to the lateral LPV model and the outer, χ, loop design starts. The outer loop performance is set up as a step tracking requirements. The ideal model is given as

$$G_{id_{lat}} = \frac{0.2}{s + 0.2}$$

Since the model is generally numerically ill conditioned, it can happen that integrators in the model appear as stable or unstable real poles with values in the order of magnitude of 10^-9. In order to overcome such numerical issues, a soft constraint for the outer lateral loop is to have all closed loop poles ≤ -0.01. Similarly as in the inner loop, in case the performances cannot be met, the design is repeated with design criteria relaxed by 10% until a feasible controller is found.

5.1.3 Inner longitudinal loop design

The inner longitudinal loop takes the LPV model containing the longitudinal states, inputs and outputs. The performance is set up as a step tracking requirement. The ideal model is given as

$$G_{id_{long}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $\omega_n = 1/3 \times \omega_{n_{short\,period}}$ is the natural frequency and $\zeta = 0.8$ is the damping factor of the ideal model. $\omega_{n_{short\,period}}$ is the short period frequency of the longitudinal model. In addition to the tracking requirement, $5dB$ gain and $40°$ phase margins are required as well. If the performances cannot be met, the design is repeated with design criteria relaxed by 10% until a feasible controller is found.

5.1.4 Outer longitudinal loop design

Before the outer, h, loop design is started, the inner loop controller is connected to the longitudinal model. Time domain control performance specifications are given as response time of $3s$, steady state error of 0%, peak error as 1.2. Since the model is generally numerically ill conditioned, it can happen that integrators in the model appear as stable or unstable real poles with values in the order of magnitude of 10^-9. In order to overcome such numerical issues, a soft constraint for the outer longitudinal loop is to have all closed loop poles ≤ -0.01. Similarly, these conditions are automatically relaxed in case a feasible controller is not found.

5.1.5 Sideslip loop design

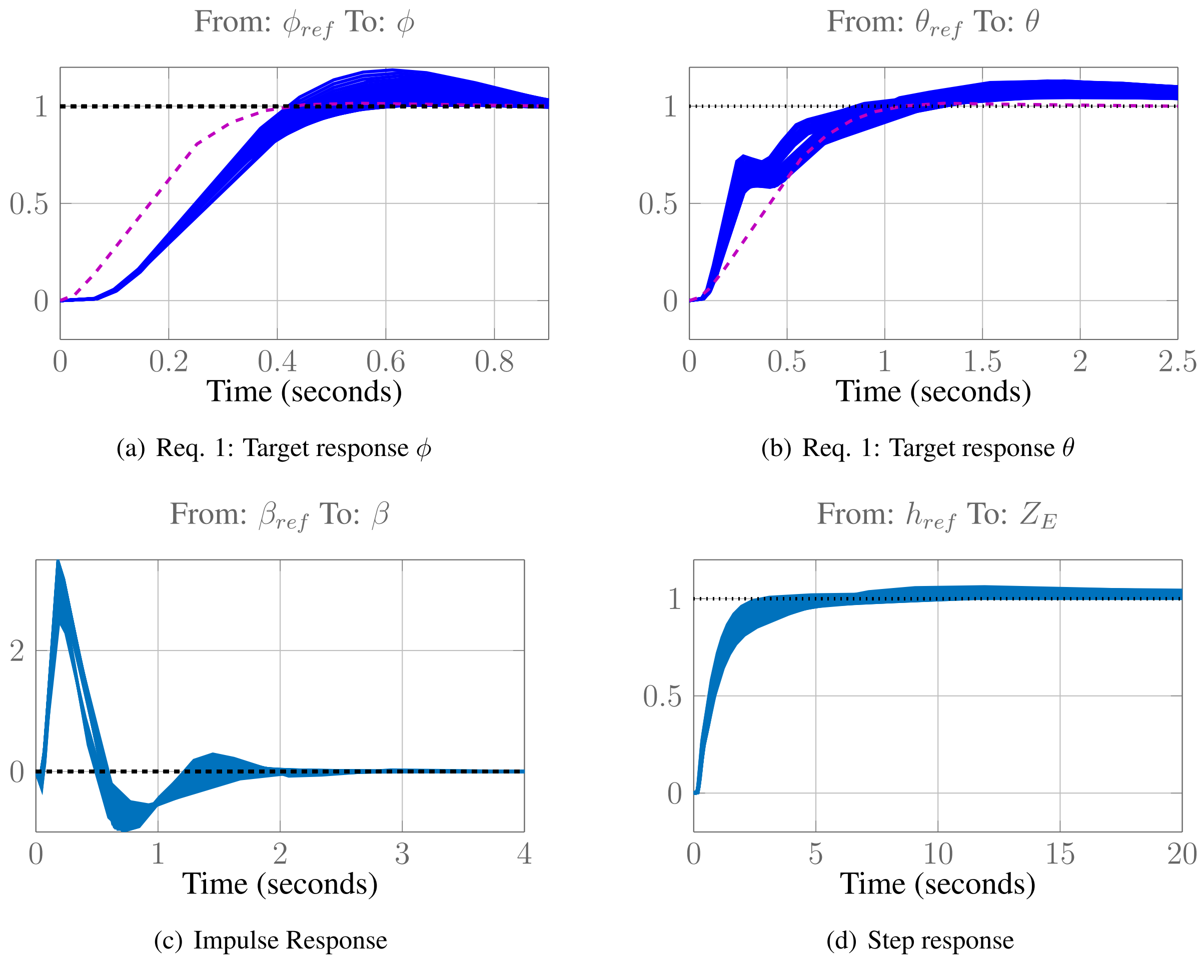
The sideslip, β, loop takes the LPV model connected with the inner and outer controllers. This ensures that the sideslip controller to be designed does not interfere with the lateral and longitudinal controllers. Since the main goal of this loop is to keep $\beta = 0$, the performance is set up as a disturbance rejection for a frequency range between 0 and $1\,rad/s$ and the damping of the closed loop is set to be $\zeta_{cl} > 0.125$. In addition, $5dB$ gain and $40°$ phase margins are required as well.

5.1.6 Airspeed loop design

For the airspeed, V_{IAS}, loop the 12 state LPV model is connected with the lateral, longitudinal and sideslip controllers. The performance is set as tracking requirement with response time of $35s$, steady state error of 0, peak error as 1.5. Some of the design requirements and performances of the baseline control design can be seen in Figure 13.

Finally, the 4 controllers are connected to the 12 state rigid body model. The response can be seen in Figure 14. Further details about the baseline control design can be found in [22, 23].

The airspeed dependent PID gains for all baseline loops are are saved in the ToolSpecific section of CPACS under the name Baseline. These are the main outputs of the block. A simple metric is also returned for the user which indicates the performance of the control loops. This allows the interaction with the automated design process: the user can formulate tighter or loser specifications according to the individual needs. A clear graphical representation is also provided which can be included in the reporting.



(a) Req. 1: Target response ϕ

(b) Req. 1: Target response θ

(c) Impulse Response

(d) Step response

Figure 13: Ezxample baseline control design requirements and results

5.2 Flutter control design functions (SZTAKI)

The flutter controller design is done based on the uncertain LPV ASE model of the aircraft. Since rigid body modes might be unstable and the aim of the flutter controller is to stabilize the flutter modes, the inner loops of the baseline controller are connected to the model. The flutter control design block takes the outer aileron (denoted by L4 and R4) actuator dynamics and the flutter control design model FlexACModel as inputs via CPACS.

The internal structure of the flutter controller is shown in Figure 15. The flexible model of the aircraft is split into the lateral and longitudinal model. The two flutter modes appear separated in these two system, hence the this separation allows us to stabilize them one-at-a-time. The controller designed to stabilize the symmetric (longitudinal) and asymmetric (lateral) flutter mode is denoted by K_{sym} and K_{asym} respectively. Both of these controllers are SISO and they are both augmented by a low pass filter

$$F(s) = \frac{1.6 \cdot 10^5}{s^2 + 560s + 1.6 \cdot 10^5} \tag{6}$$

to limit their bandwidth. The input of the flutter controller consists of the pitch rate (q), and angular rate measurement from the L6 and R6 IMU sensors (q_{L} and q_{R}). The actuating signals are the deflection commands for the pair of outermost ailerons ($u_{\text{f,L}}$ and $u_{\text{f,R}}$). These signals are blended together as depicted in Figure 15.

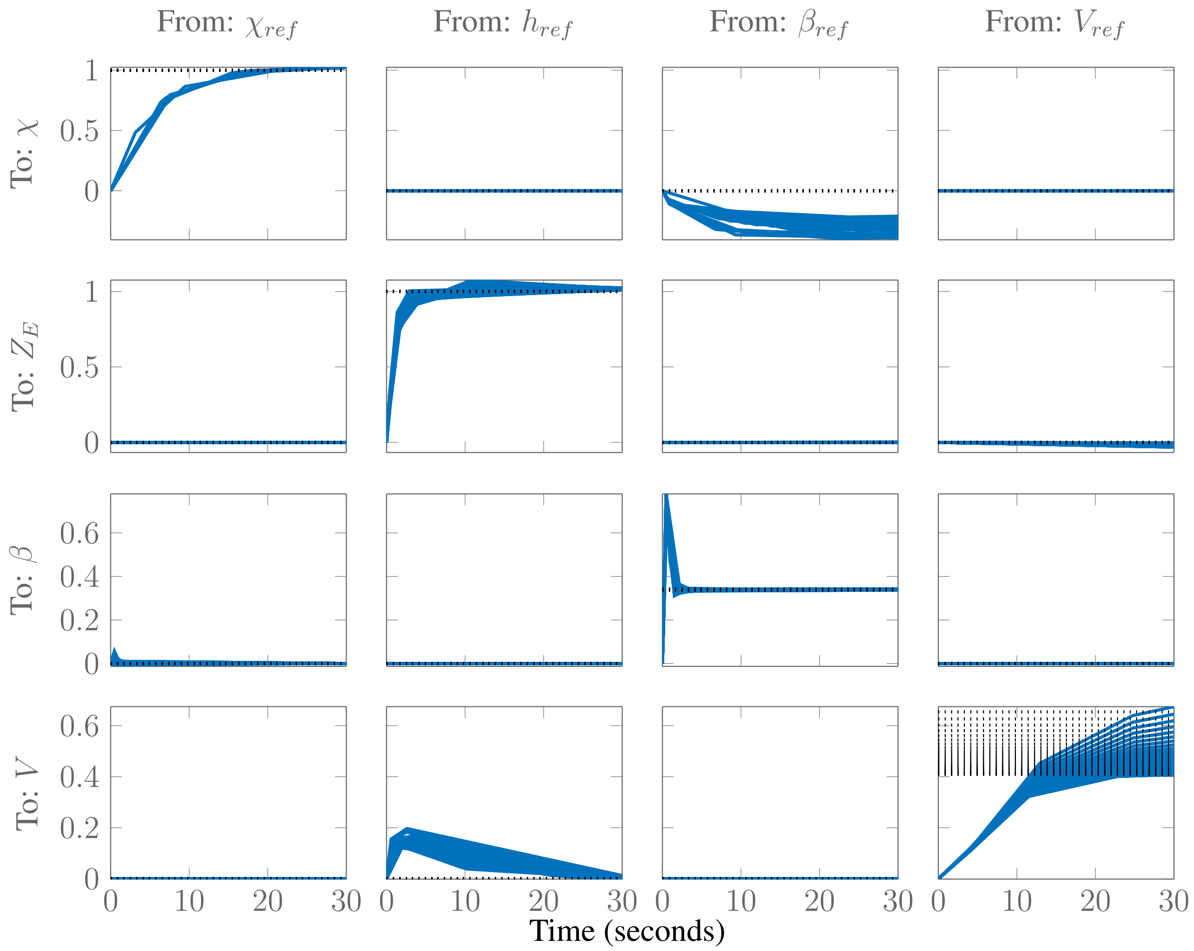


Figure 14: Baseline control step response

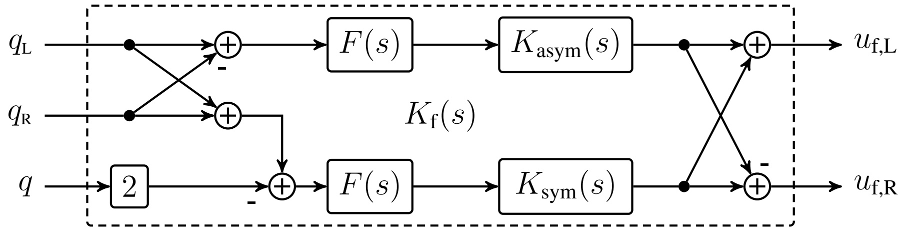


Figure 15: Internal structure of the flutter controller.

The design of both K_sym and K_asym is carried out for the nominal and uncertain values in the structural dynamics. The controllers are designed for these values simultaneously with structured H_∞ synthesis for values of the airspeed between 40m/s and 70m/s. The generalized plant interconnection is illustrated in Figure 16. Here, the weighting functions are $W_e(s) = 1/2$, $W_u(s) = 1/10°$, and $\tau(s)$ is the forth order Padé approximation of 15ms delay. The objective of the synthesis is to minimize the sensitivity function of the closed-loop which results in robust stabilization. For a detailed reasoning for and explanation of this design process see [24]. The MIMO flutter controller is then formed by the interconnection of SISO controllers and the filters as in Figure 15.

Similarly to the baseline control design algorithm, the flutter suppression control design block needs to be augmented with basic analysis algorithms to verify if the resulting controller satisfies the control performance specifications. As a main measure, the multi-input multi-output (MIMO) disc margins are selected. The state-space model of the resulting flutter suppression

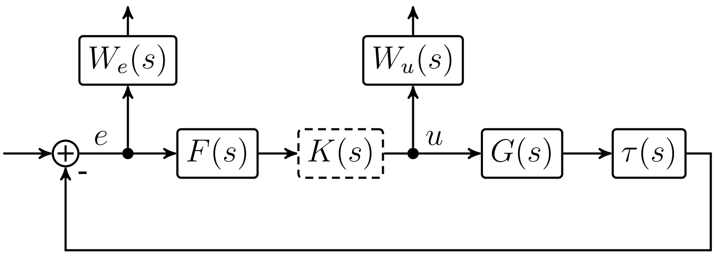


Figure 16: Generalized plant interconnection used for the flutter control design.

controller is the output of the block. The controller is saved in the ToolSpecific section of CPACS under the name Flutter.

5.2.1 RCE with Modeling and Control design blocks

The performance evaluation is done in two steps. First, it is critical to evaluate the RCE implementation of the modeling and control design blocks for the baseline and flutter suppression control design. First, an 'Input Provider' is used to send the initial CPACS file, then the Modeling component start processing and sets an output based on the actual modelling script. The output is forwarded to the Baseline (RigACModel) and Flutter (FlexACmodel) Controller components. Once the baseline design is finished, the baseline design gains are sent to the flutter design block. Finally, the Analysis block receives the flexible model, the baseline gains and the flutter controller. The modeling, he control design components and the analysis tool function with a help from external scripts which act like wrappers between them and the actual Matlab files. The scheduling between the blocks is based on the data that is the output of the preceding block. The output is set using the post-execution commands of the modeling block. The output is written in the output directory in accordance with the wrapper, so when the current block finishes, the post execution commands are executed. All RCE block communications and data sharing needs to be specified in addition to the scheduling of the RCE blocks. The control oriented modeling blocks output files are referenced in CPACS. These output files are given in the ToolSpecific field of the CPACS xml file.

The second step is to evaluate the results of the control design blocks. This step is carried out for each controller individually first. For the baseline controller the first step is to evaluate if the handling qualities are satisfied or not. If this can not be achieved by the resulting controllers then the handling qualities need to be relaxed. In addition to the handling qualities, robustness, gain and phase margins of the resulting controller is evaluated. The analysis results are also written in the corresponding ToolSpecific field of the CPACS xml file. The flutter controller is also analyzed if it satisfies the robustness analysis criteria.

5.3 Analysis of the designed controllers (SZTAKI)

The final block of the MDO toolchain is the analysis block. In the present case, the performance of the baseline and flutter controllers are evaluated. Figure 17 shows the disk margins obtained by this analysis.

For the closed loop analysis the baseline and flutter controllers are connected with the flexible aircraft model. The analysis of the closed-loop is based on disk margin calculations. Complex scalar uncertainties are injected into the channels involved in the feedback loops and the phase and gain combination at which the closed-loop becomes unstable is computed in each channel, simultaneously. First, the robustness of the baseline controller is analyzed without the flutter controller. The speed at which the disk margins become zero is considered the open-loop flutter

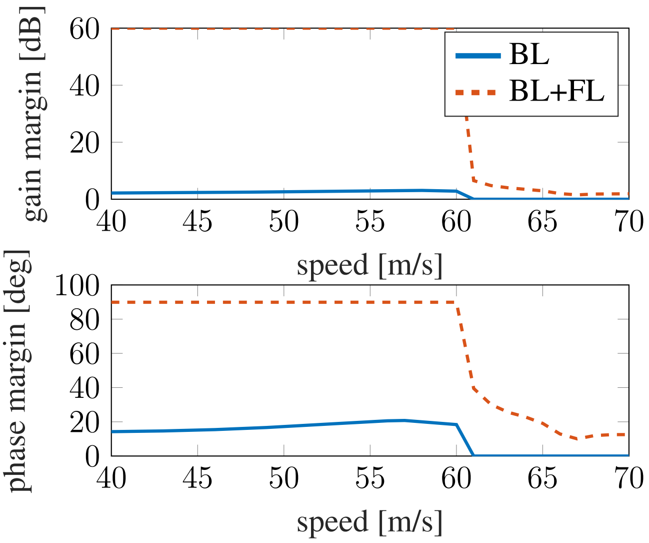


Figure 17: Resulting disk margins

speed. In the next step, the flutter controller is also connected to the system and the margins are recaclulated. This step reveals how much the flutter controller is able to extend the safe flight envelope functioning simultaneously with the baseline controller. For a more detailed explanation the reader is again referred to [24]. The analysis block determines the robust flutter speed. This is the lowest speed where the closed loop gain margin falls below 1.5 dB or the gain margin falls below 15 degrees. The absolute flutter speed is the lowest airspeed at which the flutter mode becomes unstable. The output of the analysis block are the open loop, the absolute and robust flutter speeds.

6 RESULTS OF THE PARAMETER STUDY IN THE MDO CO-DESIGN TOOLCHAIN

A parameter study for the T-Flex demonstrator was set up in the co-design MDO toolchain. The design variables are the wing sweep angle Λ and the flutter mass m_{fl}. The design values are given in Tables 1 and 2. The MDO setup is set up based on the configuration presented in the paper. All parameter cases run through in RCE with one study taking around one hour computational time.

Table 1: Sweep angle values

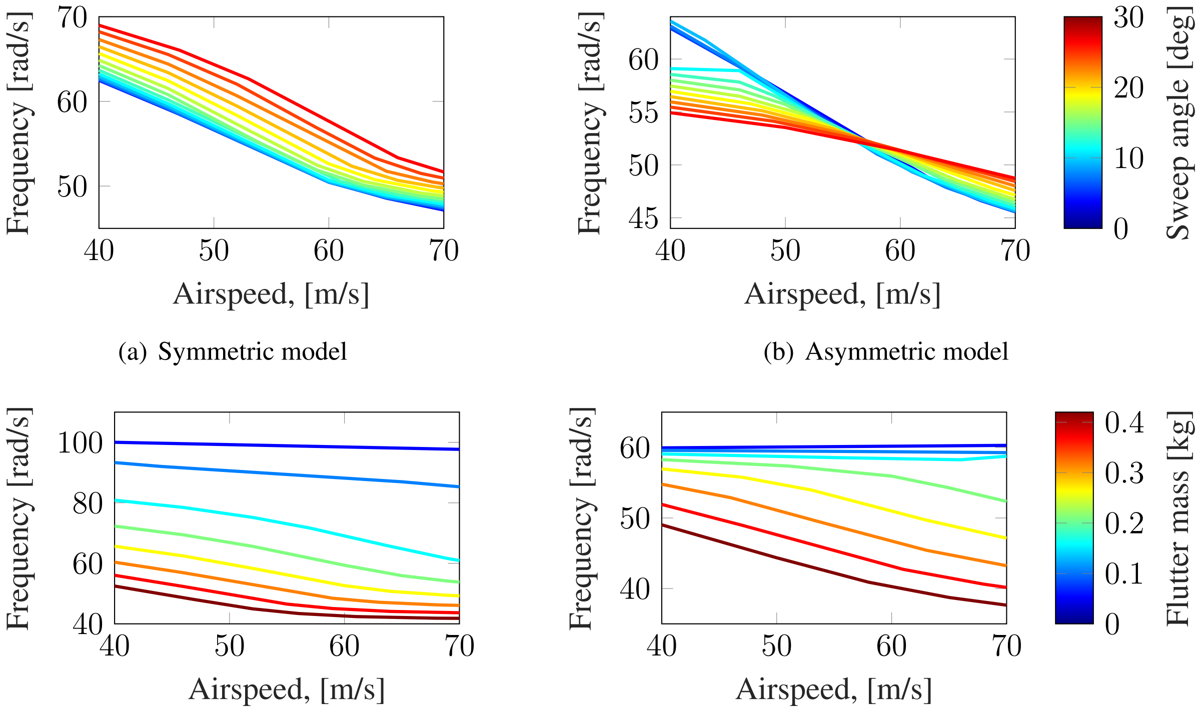
0	2.5°	5°	7.5°	10°	12.5°	15°	17.5°	20°	22.5°	25°	27.5°	30°

Table 2: Flutter mass values

$0\,kg$	$0.06\,kg$	$0.12\,kg$	$0.18\,kg$	$0.24\,kg$	$0.3\,kg$	$0.36\,kg$	$0.42\,kg$

First, the open loop results are evaluated. For this the symmetric and asymmetric flutter modes are assessed separately. Figures 18 and 19 show the results of the parameter study with respect to the flutter frequency and damping as a function of airspeed. It can be observed that the sweep angle makes both the symmetric and asymmetric flutter mode frequencies highly airspeed dependent. This could potentially indicate a more difficult active flutter suppression design. In case of the symmetric flutter mode, the flutter frequency increases as the sweep angle increases, requiring higher actuator bandwidth and higher sampling rate for the flutter suppression controller. The flutter mass, as expected, has a huge impact on the flutter frequency.

The damping plots show the airspeed where the flutter modes become unstable for each param-



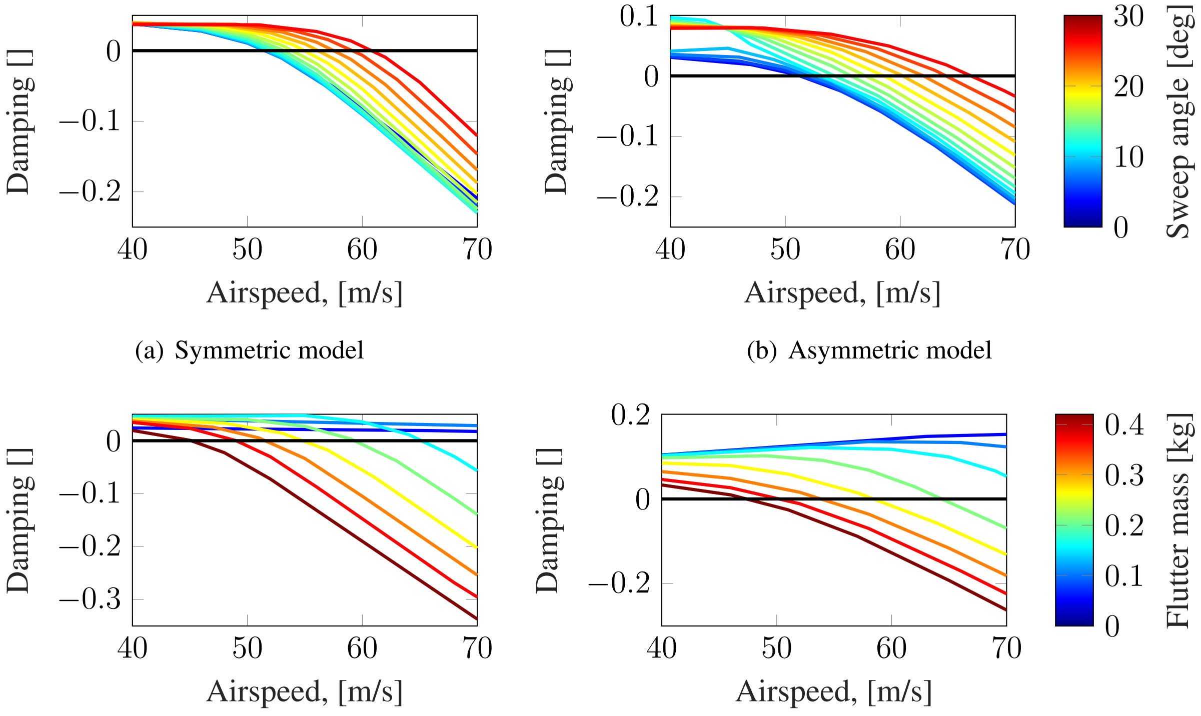
(a) Symmetric model

(b) Asymmetric model

(c) Symmetric model

(d) Asymmetric model

Figure 18: Flutter frequency with respect to the airspeed and parameter variation



(a) Symmetric model

(b) Asymmetric model

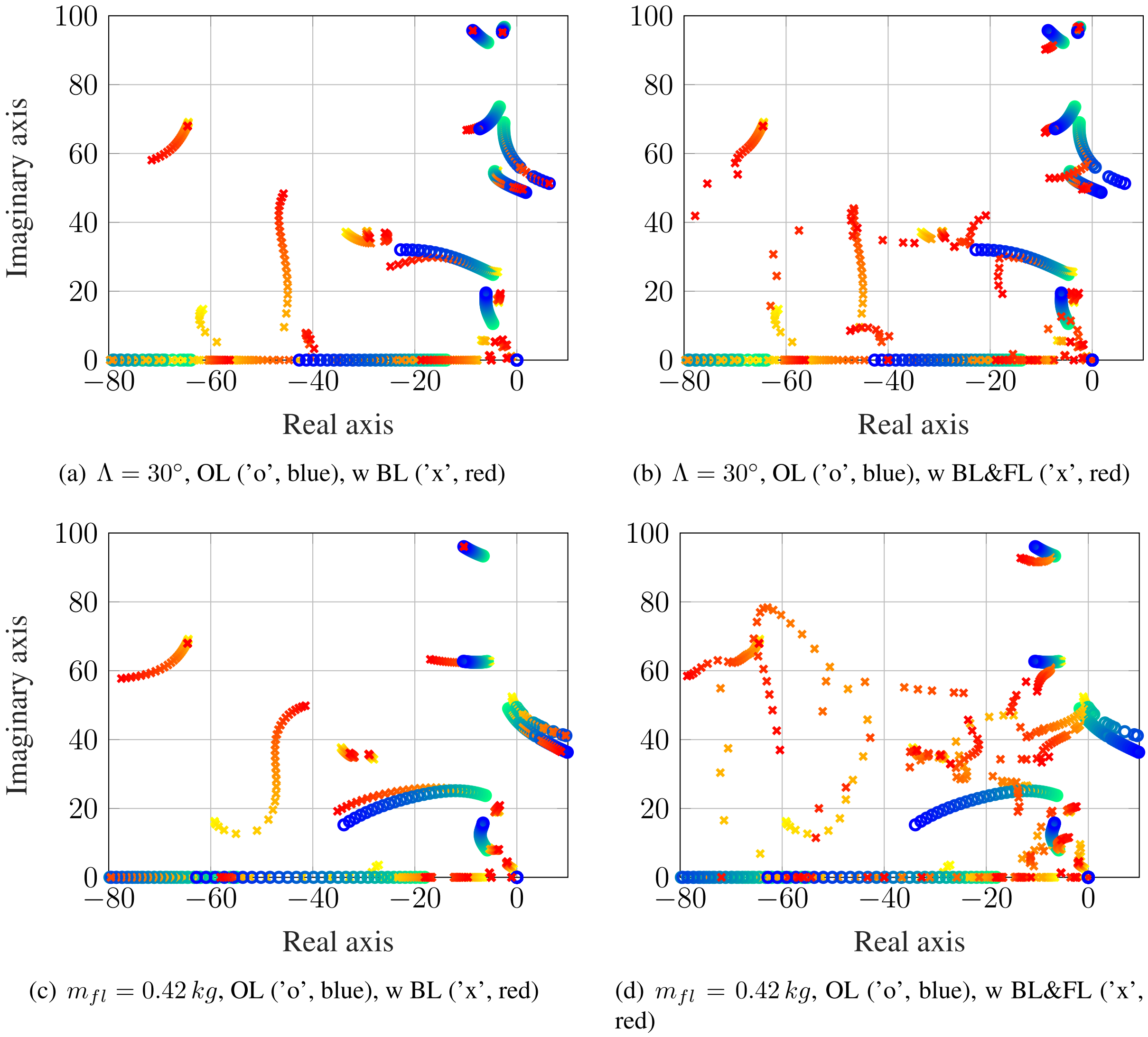
(c) Symmetric model

(d) Asymmetric model

Figure 19: Flutter damping with respect to the airspeed and parameter variation

eter study case. This happens when the damping becomes negative. It can be observed that the higher sweep angles push the open loop flutter speed higher. Once the flutter modes become unstable the slopes of the damping curves remain similar. This means that the sweep angle variation does make a huge impact on how aggressive the flutter mode is. In case of the flutter mass variation, low flutter mass values result in the flutter modes not going unstable up until 70 m/s airspeed. The higher the flutter mass is, the lower the open loop flutter speed becomes.

Second, the closed loop performance is evaluated. Figure 20 shows example of the pole migrations for $\Lambda = 3°$ and $m_{fl} = 0.42\,kg$.



(a) $\Lambda = 30°$, OL ('o', blue), w BL ('x', red)

(b) $\Lambda = 30°$, OL ('o', blue), w BL&FL ('x', red)

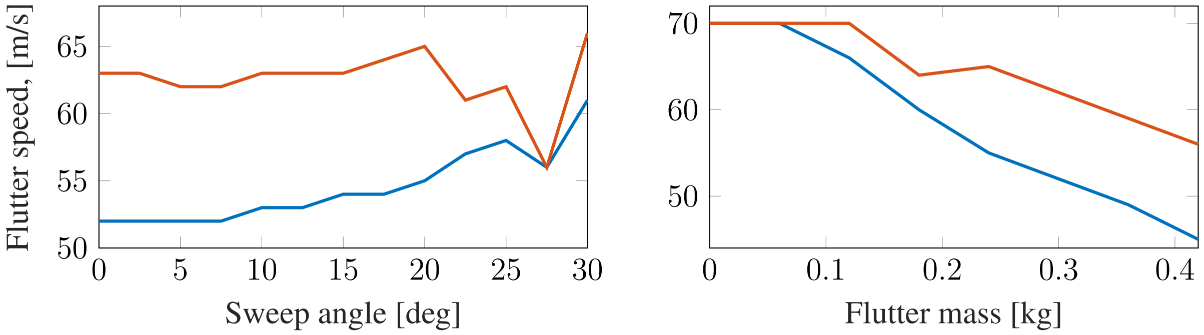
(c) $m_{fl} = 0.42\,kg$, OL ('o', blue), w BL ('x', red)

(d) $m_{fl} = 0.42\,kg$, OL ('o', blue), w BL&FL ('x', red)

Figure 20: Open ('o', blue) and closed loop ('x', red) pole migration examples

In Figures 20.a and 20.c the pole migrations of the open loop model and the model with the baseline controller are depicted. It can be observed that the flutter modes are only very mildly affected by the baseline controller. In Figures 20.b and 20.d the flutter controller is also added on top of the baseline controller. The flutter modes are pushed to the left, the baseline and flutter controllers are operating as expected.

Figure 21. shows the open loop and robust closed loop flutter speeds with respect to the parameter study. As in Figure 18., the open loop flutter speed increases by increasing the sweep angle. The closed loop flutter speed remains at a relatively constant level for low sweep angles after which it start to decrease. At $\Lambda = 27.5°$ the open loop robust flutter speed are the same. This, upon a detailed investigation, turns out to be due to the baseline controller. Recall, that the baseline controller is designed based on the 12 state model. Once it is connected with the flexible model, for this specific case the baseline controller pushes a different aeroelastic mode unstable for a certain airspeed range, which the flutter controller is not able to stabilize. It does stabilize the original flutter modes though. Retuning the baseline controller design algorithms could overcome this problem. However, this phenomena also indicates that designing flutter suppression controllers is more difficult for higher weep angles. In case of the flutter mass variation, the open loop and closed loop flutter speeds are at 70 m/s airspeed, which is the limit until which the MDO was run. At higher masses, the open loop flutter speed decreases, but the relative gain in the closed loop flutter speeds becomes higher.



(a) Sweep angle variation

(b) Flutter mass variation

Figure 21: Open loop (blue) and closed loop robust (red) flutter speed variations

Third, the mission analysis is evaluated. The effect of the sweep angle and flutter mass on the open loop, robust closed loop and absolute closed loop flutter speeds is summarized in Table 3.

Table 3: RCE results; red: reference configuration, blue: best Λ case, violet: best m_{fm} case

Mass [kg]	Sweep [deg]	Open loop flutter [m/s]	Robust flutter [m/s]	Increase [%]
0.24	0	52	63	21.15
0.24	2.5	52	63	21.15
0.24	5	52	62	19.23
0.24	7.5	52	62	19.23
0.24	10	53	63	18.87
0.24	12.5	53	63	18.87
0.24	15	54	63	16.67
0.24	17.5	54	64	18.52
0.24	20	55	65	18.18
0.24	22.5	57	61	7.02
0.24	25	58	62	6.9
0.24	27.5	56	56	0
0.24	30	61	66	8.2
0	20	70	70	0
0.06	20	70	70	0
0.12	20	66	70	6.06
0.18	20	60	64	6.67
0.3	20	52	62	19.23
0.36	20	49	59	20.41
0.42	20	45	56	21.74

It can be seen that the lowest sweep angles give around 21% increase in the closed loop flutter speed, while the increase in case of the highest flutter mass is almost 22%. Note, that a higher granularity in the airspeed resolution would probably lead to $\Lambda = 0$ as the optimal parameter for the sweep angle. Both of the optimal parameters individually resulted in more than 3% increase with respect to the reference aircraft model.

7 CONCLUSIONS

Future aircraft designs will result in airframes with increased structural flexibility. This increased flexibility requires additional control loops, for example active flutter suppression controller. In a classical approach, the conceptual airframe is a result of an MDO process where the structure and the aerodynamics are optimized in a single step. The control design comes later in the design process. Since flexible aircraft require increased complexity in the control design, the classical MDO might result in a sub-optimal airframe design. The paper proposes an extended MDO for flexible aircraft, called co-design, where the airframe parameters and the controllers can be otimizied in a single step. The proposed co-design is presented through the T-Flex demonstrator aircraft and the implementation is done in the RCE framework. In order to achieve the proposed co-design, the modelling and control design algorithms have to be set up in a way that can run automatically without human interaction. The control design algorithms of the co-design considered in the paper are the baseline and the flutter suppression controllers. The design parameters of the aircraft are chosen to be the sweep angle and the flutter mass. The objective function of the mission analysis is the relative increase of the flutter speed with active suppression. The lowest sweep angles and the highest flutter mass values resulted in the optimal airframe configuration, with above 3% increase over the reference airframe performance. The presented parameter study demonstrates the applicability and the possible benefits of the proposed co-design approach for conceptual aircraft design.

In the present paper the sweep angle or the flutter mass were varied only one at a time. In addition, the workflow is set up to handle the ply angle variations as well. As a future work, we plan to run the study over the full grid of the available design parameters to get richer data.

ACKNOWLEDGMENTS

The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002).

The research was supported by the ESA research project Transient and Microvibrations Control for Optical Missions in Combination with RF Instrument (ITT No. AO/1-11980/23/NL/MGu).

8 REFERENCES

[1] PAAW (2014-2019). Performance Adaptive Aeroelastic Wing Program. Supported by NASA NRA ”Lightweight Adaptive Aeroelastic Wing for Enhanced Perfromace Across the Flight Envelope”.

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

[3] FliPASED (2019-2022). Flight Phase Adaptive Aero-Servo-Elastic Aircraft Design Methods (FliPASED). Project of the European Union, Project ID: 815058.

[4] Papalambros, P. Y. and Wilde, D. J. (2017). *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press, 3 ed. doi:10.1017/9781316451038.

[5] Mocsányi, R. D., Takarics, B., Kotikalpudi, A., et al. (2020). Grid-based and polytopic linear parameter-varying modeling of aeroelastic aircraft with parametric control surface design. *Fluids*, 5(2), 47. doi:10.3390/fluids5020047.

[6] Van, E. N., Alazard, D., Döll, C., et al. (2019). Co-design of aircraft vertical tail and control laws using distributed electric propulsion. *IFAC-PapersOnLine*, 52(12), 514–519. doi:10.1016/j.ifacol.2019.11.295.

[7] Denieul, Y., Bordeneuve, J., Alazard, D., et al. (2018). Multicontrol surface optimization for blended wing–body under handling quality constraints. *Journal of Aircraft*, 55(2), 638–651. doi:10.2514/1.c034268.

[8] Moreno, C., Gupta, A., Pfifer, H., et al. (2014). Structural model identification of a small flexible aircraft. In *American Control Conference*. pp. 4379–4384.

[9] Kotikalpudi, A. (2017). *Robust Flutter Analysis for Aeroservoelastic Systems*. Ph.D. thesis, University of Minnesota, Twin Cities.

[10] Schmidt, D. K., Zhao, W., and Kapania, R. K. (2016). Flight-dynamics and flutter modeling and analysis of a flexible flying-wing drone. In *AIAA Atmospheric Flight Mechanics Conference, AIAA SciTech Forum*.

[11] Meddaikar, Y. M., Dillinger, J., Klimmek, T., et al. (2019). Aircraft aeroservoelastic modelling of the FLEXOP unmanned flying demonstrator. In *AIAA Scitech 2019 Forum*. AIAA.

[12] Takarics, B., Vanek, B., Kotikalpudi, A., et al. (2018). Flight control oriented bottom-up nonlinear modeling of aeroelastic vehicles. In *2018 IEEE Aerospace Conference*. IEEE.

[13] Shamma, J. S. (1988). *Analysis and design of gain scheduled control systems*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge.

[14] Becker, G. (1993). *Quadratic stability and performance of linear parameter dependent systems*. Ph.D. thesis, University of California, Berkeley.

[15] Boden, B., Flink, J., Mischke, R., et al. (2019). Distributed Multidisciplinary Optimization and Collaborative Process Development Using RCE. In *AIAA Aviation 2019 Forum, 17–21 June 2019, Dallas, TX, USA*. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2019-2989.

[16] Roessler, C., Stahl, P., Sendner, F., et al. (2019). Aircraft design and testing of FLEXOP unmanned flying demonstrator to test load alleviation and flutter suppression of high aspect ratio flexible wings. In *AIAA Scitech 2019 Forum*. AIAA.

[17] Boden, B., Flink, J., Först, N., et al. (2021). RCE: An integration environment for engineering and science. *SoftwareX*, 15, 100759. doi:10.1016/j.softx.2021.100759.

[18] Flink, J., Mischke, R., Schaffert, K., et al. (2022). Orchestrating tool chains for model-based systems engineering with RCE. In *2022 IEEE Aerospace Conference (AERO)*. IEEE. doi:10.1109/aero53065.2022.9843838.

[19] Wuestenhagen, M., Kier, T., Meddaikar, Y. M., et al. (2018). Aeroservoelastic modeling and analysis of a highly flexible flutter demonstrator. In *2018 Atmospheric Flight Mechanics Conference*. AIAA.

[20] Wu, F. (1995). *Control of Linear Parameter Varying Systems*. Ph.D. thesis, Univ. California, Berkeley.

[21] Vinnicombe, G. (1993). *Measuring Robustness of Feedback Systems*. Ph.D. thesis, Univ. Cambridge, Cambridge.

[22] Luspay, T., Ossmann, D., Wuestenhagen, M., et al. (2019). Flight control design for a highly flexible flutter demonstrator. In *AIAA Scitech 2019 Forum*. AIAA.

[23] Takarics, B., Patartics, B., Luspay, T., et al. (2023). Model based automatic control design for the t-flex demonstrator using rce environment. In *AIAA SCITECH 2023 Forum*. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2023-0175.

[24] Patartics, B., Lipták, G., Luspay, T., et al. (2021). Application of structured robust synthesis for flexible aircraft flutter suppression. *IEEE Transactions on Control Systems Technology*, 30(1), 311–325.

COPYRIGHT STATEMENT