

The following report was submitted for consideration to the IEEE Transactions on Control Systems Technology as a regular paper on the 2nd of March 2020 and is under review as of the 17th of July 2020. The authors emphasize that this document is not a peer reviewed publication.

Application of Structured Robust Synthesis for Flexible Aircraft Flutter Suppression

Bálint Patartics, György Lipták, Tamás Luspay, Peter Seiler, *Member, IEEE*, Béla Takarics, *Member, IEEE*, and Bálint Vanek, *Member, IEEE*,

Abstract—This paper presents a method for structured robust control design for systems with a mixture of parametric and dynamic uncertainty. The proposed method alternates between an analysis step and a synthesis step. Samples of the parametric uncertainty are computed during the analysis steps thus yielding an array of uncertain systems containing only dynamic uncertainty. The controller is then synthesized on this array of uncertain models. This synthesis step itself involves an alternation between constructing a D-scale for each of the uncertain systems and tuning a single controller for the entire collection of scaled plants. The controller tuning is performed using structured control design techniques. The proposed method is utilized to design a flutter suppression controller for a flexible aircraft. The aircraft dynamics are described by both a high fidelity and a reduced-order model. The design objectives for flutter suppression are to achieve robust stabilization in the presence of mixed uncertainty. The proposed structured design method yields a single, low order, linear time invariant controller which increases the flutter speed by 15%. Additional robustness analyses and high fidelity simulations are provided to assess the controller performance.

Index Terms—uncertain systems, structured synthesis, flutter control

I. INTRODUCTION

THERE are a variety of optimal control synthesis methods available in the literature, e.g. H_2 and H_∞ methods [1], [2]. However, the practical application of these methods can be limited because they provide full-structure state-space controllers. Industrial applications often prefer or require controllers with specific structure for implementation. Examples include the VEGA launch vehicle [3], automated landing of a civilian aircraft [4], and chemical process control [5].

The H_∞ synthesis conditions can be expressed as convex constraints when the plant is nominal (no uncertainty) and the controller is allowed to be full-order and unstructured [6]. Nominal synthesis becomes non-convex, in most cases, once structure or reduced order is imposed on the controller. One notable exception is the quadratic invariance condition [7] which allows for convex synthesis. As a consequence, most

research on structured control synthesis has focused on algorithms that are reliable and computationally efficient on typical engineering problems (but not necessarily with proofs of global optimality).

Two notable algorithms for structured control synthesis include the H_∞/H_2 Fixed Order Optimization (HIFOO) in [8] and structured H_∞ design in [9]. This paper focuses on the algorithm in [9] due to its ease of use in MATLAB [10] as the `hinfstruct` function. The approach has also been extended to handle multiple frequency domain design requirements in the fixed structured design as implemented in the `systune` function [11].

This paper builds on a previous conference paper [12] in proposing a structured synthesis method for systems with mixed (i.e. parametric and dynamic) uncertainty. The chosen objective of the design is the minimization of the worst-case gain, i.e. the maximal closed-loop H_∞ norm over the set of allowable uncertainties. The algorithm utilizes an iteration in which controller synthesis is followed by analysis. This iteration continuously updates a set of “bad” samples for the parametric uncertainty. This yields a corresponding set of sampled design plants. The synthesis step itself is an iterative process, which resembles the traditional D-K iteration [13]. The K-step is performed as a structured design (using `systune`) on the collection of sampled plants. The D-step solves for the scalings associated with each sampled plant. In the analysis step, the worst-case gain of the closed loop is calculated along with a worst-case uncertainty. A sample of parametric uncertainty is extracted from this worst-case uncertainty. The subsequent synthesis step is repeated using this updated set of sampled plants.

The proposed algorithm resembles the hybrid relaxation approach in [14, Algorithm 2]. The novelty of the paper lies in the control synthesis step. In particular, the hybrid relaxation given in [14] has a synthesis step that involves parameterizing a single dynamic D-scale for all uncertainty sample. It then jointly optimizes over the tunable parameters in both the scaling and controller using structured synthesis techniques [9], [10]. This approach is effective for many problems but the restriction to a single dynamic scaling might be conservative on certain examples, e.g. see Section IV-B in [12]. As commented on in [14], it is possible to extend the hybrid relaxation to parameterize one scaling for each parametric uncertainty sample at the expense of a larger number of optimization variables.

In contrast, the approach given in this paper utilizes dynamic scalings that depend on the parametric uncertainty sample. The

Manuscript created February 02, 2020.

B. Patartics, T. Luspay, B. Takarics, and B. Vanek are and Gy. Lipták was with the Institute for Computer Science and Control, Kende u. 13-17, H-1111 Budapest, Hungary (emails: patartics.balint@sztaki.hu, luspay.tamas@sztaki.hu, takarics.bela@sztaki.hu, vanek@sztaki.hu, and liptak@sztaki.mta.hu)

P. Seiler was with the Aerospace Engineering and Mechanics Department, University of Minnesota, 107 Akerman Hall, 110 Union St. SE, Minneapolis MN 55455, USA (email: seile017@umn.edu)

The research leading to these results is part of the FLEXOP project. This project has received funding from the Horizon 2020 research and innovation programme of the European Union under grant agreement No 636307.

use of parameter-dependent scalings reduces conservatism at the expense of requiring a coordinate-wise D-K iteration for the synthesis. This coordinate-wise iteration alternates between optimizing the D-scaling while holding the controller fixed and vice versa. This tends to be less effective than the joint scaling and controller optimization used for the synthesis in the hybrid relaxation approach. If the D-scales depend strongly on the value of the uncertain parameters however, the individual D-scale construction has the advantage. This is illustrated with a concrete example in [12]. Another minor distinction is that we use a branch-and-bound implementation for the worst-case gain analysis. The branch-and-bounding provides tighter analysis bounds. MATLAB implementation of the algorithm is available on-line [15].

To demonstrate the effectiveness of this synthesis method, and as an extension of our previous work in [12], a real-life flutter suppression control example is presented for the demonstrator aircraft of the FLEXOP project [16]. The aeroelastic flutter is an adverse interaction between the aerodynamics and the structural dynamics of the aircraft which causes undamped oscillations that can result in catastrophic structural failure. There are a number of active control solutions to mitigate flutter. The robust control framework was successfully applied in [17] and extended to the linear parameter varying case in [18]. Optimal blending of the inputs and outputs is utilized in [19] to maximize control effectiveness for the flutter modes and minimize the excitation of the remaining modes.

For our approach, an uncertain control-oriented model of the aircraft is obtained with both parametric and dynamic uncertainty. The control problem is decomposed into the design of two independent SISO controllers with the performance criteria of robust stabilization and control effort minimization. The robustness of the flutter control loop is evaluated using the high fidelity model of the aircraft. Loop margins are computed and time domain simulations are conducted to investigate the extension of the safe flight envelope.

The remainder of the paper is laid out as follows. The synthesis method is described in detail by Section II. Section III presents the nonlinear high fidelity model of the aircraft, the bottom-up modeling that results in the control oriented model, the construction of the uncertain model, and the control design. The analysis of the closed flutter loop is given in Section IV. Finally, conclusions are drawn in Section V.

II. STRUCTURED ROBUST CONTROL DESIGN ALGORITHM AGAINST MIXED UNCERTAINTY

The majority of this section is the restatement of the method in [12]. The mathematical formulation of the design problem is given in Section II-A. Then, Section II-B provides an overview of the algorithm. The construction of the D-scales, and the synthesis and analysis steps are elaborated in Sections II-C, II-D, and II-E respectively.

A. Problem formulation

1) *Notation* : Let $N \in \mathbb{C}^{r_N \times c_N}$ and $K \in \mathbb{C}^{r_K \times c_K}$ be given matrices with $r_K < c_N$ and $c_K < r_N$. Partition N such that

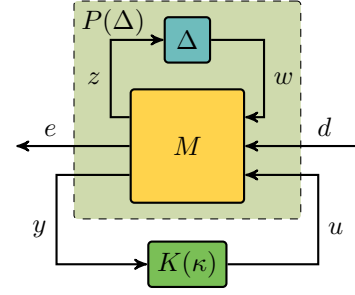


Fig. 1. Closed loop interconnection for the control design.

the lower right block is $c_K \times r_K$, and

$$N = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}. \quad (1)$$

The lower LFT (Linear Fractional Transformation) is

$$\mathcal{F}_L(N, K) := N_{11} + N_{12}K(I - N_{22}K)^{-1}N_{21}. \quad (2)$$

Similarly, the upper LFT of N and $\Delta \in \mathbb{C}^{r_\Delta \times c_\Delta}$ is defined as

$$\mathcal{F}_U(N, \Delta) := N_{22} + N_{21}\Delta(I - N_{11}\Delta)^{-1}N_{12}, \quad (3)$$

assuming compatible dimensions. The same definitions for both the lower and upper LFTs hold if N , K and, Δ are linear time-invariant systems. Finally, $\text{diag}\{\Delta_1, \Delta_2, \dots, \Delta_N\}$ denotes the block diagonal concatenation of $\Delta_1, \Delta_2, \dots, \Delta_N$.

2) *Structured Robust Synthesis* : The robust synthesis problem is formulated using the feedback interconnection shown in Fig. 1. The uncertain plant $P(\Delta) = \mathcal{F}_U(M, \Delta)$ consists of an LTI system M and structured LTI uncertainty Δ . The objective is to design an LTI controller with fixed structure, $K(\kappa)$, where $\kappa \in \mathbb{R}^{n_\kappa}$ is a vector of tunable design parameters.

The uncertainty Δ is assumed to be block-structured consisting of unit norm-bounded parametric and dynamic uncertainty [20], [21]. Specifically, define the following sets of parametric and dynamic uncertainty

$$\begin{aligned} \mathbb{A}_p &:= \left\{ \text{diag} \left\{ \delta_1 I_{r_1}, \dots, \delta_{N_p} I_{r_{N_p}} \right\} \mid \delta_i \in \mathbb{R}, |\delta_i| \leq 1 \right\} \\ \mathbb{A}_d &:= \left\{ \text{diag} \{ \Delta_1, \dots, \Delta_{N_d} \} \mid \Delta_i \text{ LTI}, \|\Delta_i\|_\infty \leq 1 \right\}, \end{aligned} \quad (4)$$

where $\|\cdot\|_\infty$ denotes H_∞ norm. The dynamic blocks in \mathbb{A}_d are assumed to be square. This assumption can be relaxed with only notational changes. The plant uncertainty is given as $\Delta \in \mathbb{A}$ where

$$\mathbb{A} := \{ \text{diag} \{ \Delta_p, \Delta_d \} \mid \Delta_p \in \mathbb{A}_p \text{ and } \Delta_d \in \mathbb{A}_d \}. \quad (5)$$

Note that $\|\Delta\|_\infty \leq 1$ for all $\Delta \in \mathbb{A}$. We also define a set of D-scales \mathbb{D}_d associated with the set of dynamic uncertainties \mathbb{A}_d . The set \mathbb{D}_d consists of stable, minimum phase, and invertible LTI systems which commute with the elements of \mathbb{A}_d , i.e. if $D_d \in \mathbb{D}_d$ then $D_d \Delta_d = \Delta_d D_d$, $\forall \Delta_d \in \mathbb{A}_d$. This set of scalings will be used in the structured robust synthesis algorithm.

Define the worst-case gain of the closed loop in Fig. 1 as

$$J(\kappa) := \max_{\Delta \in \mathbb{A}} \|\mathcal{F}_L(P(\Delta), K(\kappa))\|_\infty. \quad (6)$$

This definition assumes that $\mathcal{F}_L(P(\Delta), K(\kappa))$ is robustly stable, i.e. stable for all $\Delta \in \mathbb{A}$. If the closed-loop is not

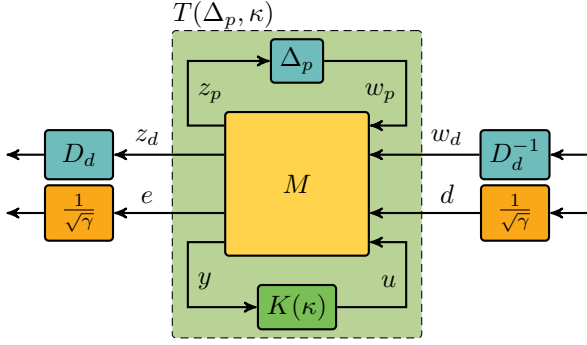


Fig. 2. Scaled plant used in the D-K iteration.

robustly stable then $J(\kappa)$ is defined to be $+\infty$. The synthesis objective is to tune the parameters of the structured controller to minimize the closed-loop worst-case gain, i.e.

$$\kappa^* = \arg \min_{\kappa \in \mathbb{R}^n} J(\kappa). \quad (7)$$

According to our engineering experience, optimizing the worst-case gain is a better performance objective than the usual robust performance [13]. If robust stability is reached, the design objective is to improve performance without extending the stability margin any further.

Structured robust synthesis is, in general, a non-convex optimization problem. Sections II-B-II-E describe an iterative method to compute sub-optimal controller parameters.

B. Overview of the algorithm

This section introduces the proposed algorithm for computing a structured controller to minimize the closed-loop worst-case gain. An overview summary of the proposed approach is given in Algorithm 1. The algorithm is iterative and utilizes sample sets for the parametric uncertainty $\Delta_{p,s}$ and D-scales $\mathbb{D}_{d,s}$. The set $\Delta_{p,s}$ is initialized with the nominal value of the real parametric uncertainty (Step 2). These sets are updated throughout the iteration, as described further below. The algorithm also expects that a controller $K(\kappa_{\text{init}})$ is available for initialization. It is further assumed that this controller robustly stabilizes the system for the dynamic uncertainty (Δ_d) and the nominal value of the parametric uncertainty ($\Delta_p = 0$). In some problems the open loop is robustly stable and hence $K(\kappa_{\text{init}}) = 0$ can be used for initialization. Otherwise an initial robustly stabilizing controller can be computed with a related stability margin maximization algorithm.

The first key step of Algorithm 1 is to synthesize a structured robust controller using an iteration (Steps 4-7) that is similar to the (unstructured, full-order) D-K iteration [13]. This alternates between a D-step (Step 5) which computes a set of scalings $\mathbb{D}_{d,s}$ and a K-step (Step 6) which computes a structured robust controller $K(\kappa)$. This iteration terminates when $\bar{\gamma}_d$ does not decrease significantly compared to the previous iteration, or when the maximum number of iterations is reached. Our specific implementation terminates the D-K inner loop if $\bar{\gamma}_d$ fails to decrease by more than 1%, or the maximum number of 15 iterations is reached.

Algorithm 1 Structured robust synthesis.

- 1: **Given:** Initial controller $K(\kappa_{\text{init}})$ which is robustly stabilizing for Δ_d and $\Delta_p = 0$.
 - 2: **Initialize:** $\Delta_{p,s} := \{0\}$ and $\kappa := \kappa_{\text{init}}$.
 - 3: **while** termination criteria are not met **do**
 - 4: **while** D-K termination criteria are not met **do**
 - 5: **Parameter-Dependent D-Scales (Section II-C):**
 Form a set of closed-loops T with $K(\kappa)$ and each sample of parametric uncertainty $\Delta_p \in \Delta_{p,s}$ (Fig. 2). For each closed-loop, solve for the scaling $D_d \in \Delta_d$ to minimize the worst-case gain upper bound over the dynamic uncertainty Δ_d .
 Output: Set of dynamic scalings $\mathbb{D}_{d,s}$ and largest worst-case gain upper bound $\bar{\gamma}_d$.
 - 6: **Controller Synthesis (Section II-D):** Form scaled plants from each sample of parametric uncertainty and scaling $(\Delta_p, D_d) \in \Delta_{p,s} \times \mathbb{D}_{d,s}$ (Fig. 2). Use `sysune` to solve for a structured controller to minimize the worst-case gain on this collection of sampled, scaled systems.
 Output: Controller parameters κ and worst-case synthesis gain γ_s .
 - 7: **end while**
 - 8: **Analysis (Section II-E):** Use a modified version of `wcgain` to compute lower/upper bounds on the worst-case gain of the closed-loop with structured controller $K(\kappa)$ over the set of uncertainty Δ .
 Output: Bounds $(\gamma_a, \bar{\gamma}_a)$ on worst-case gain, and worst-case uncertainty $\Delta^* = \text{diag}\{\Delta_p^*, \Delta_d^*\}$.
 - 9: **Sample Update:** Set $\Delta_{p,s} := \Delta_{p,s} \cup \{\Delta_p^*\}$.
 - 10: **end while**
-

The D-step (Step 5) first constructs a set of closed-loops with the structured controller $K(\kappa)$ and each sample of the parametric uncertainty $\Delta_p \in \Delta_{p,s}$ obtained in Step 8. Each closed-loop sample still has the remaining dynamic uncertainty Δ_d . Worst-case gain analyses are performed to compute a scaling $D_d \in \Delta_d$ for each closed-loop with corresponding sample $\Delta_p \in \Delta_{p,s}$. Thus $\mathbb{D}_{d,s}$ has the same number of elements as $\Delta_{p,s}$. Details on this step are given in Section II-C.

The K-step (Step 6) forms a collection of scaled plants from the samples of parametric uncertainty and D-scales. Specifically, a scaled plant is constructed for each sample $(\Delta_p, D_d) \in \Delta_{p,s} \times \mathbb{D}_{d,s}$ as shown in Fig. 2. Then `sysune` is used to compute a structured controller $K(\kappa)$ to minimize the worst-case gain on this collection of sampled, scaled systems. Details on this step are provided in Section II-D.

Finally, an analysis step (Step 8) is performed on the resulting closed-loop $\mathcal{F}_L(P(\Delta), K(\kappa))$. In the analysis, the entire uncertainty Δ is considered using a version of `wcgain` with additional branch-and-bounding. This generates lower and upper bounds on the worst-case gain. It also computes a worst-case uncertainty that achieves the lower bound. The parametric block of this worst-case uncertainty is added to the sample set $\Delta_{p,s}$ (Step 9). Details on the analysis are given in Section II-E.

The algorithm outer-loop terminates (Step 3) if the worst-case synthesis gain is within the interval of the analysis bounds, i.e. $\gamma_s \in [\underline{\gamma}_a, \bar{\gamma}_a]$, and $\bar{\gamma}_a$ fails to decrease significantly (more than 1%) over the last iteration. The loop also terminates if the maximum number of iteration counts (30) is reached.

The outline of Algorithm 1 is inspired by [14, Algorithm 2]. The key difference between the two approaches is the control synthesis in Steps 4-7. Specifically, in Algorithm 1 a separate $D_d \in \mathbb{D}_{d,s}$ corresponds to the individual samples in $\Delta_{p,s}$, instead of a common D_d . This solution reduces conservatism, and decreases the number of decision variables for the structured synthesis. This comes at the expense of having to use an iterative process for the controller synthesis however. Another distinction between the two approaches is that we use branch-and-bounding for the worst-case gain computation, which results in tighter bounds.

The proposed algorithm is designed to handle mixed uncertainty. If the system only contains dynamic uncertainty ($\Delta = \Delta_d$) then only Steps 4-7 are performed. On the other hand, if the uncertainty is purely parametric ($\Delta = \Delta_p$) then the samples are LTI systems without uncertainty. In this case, Steps 4-7 are replaced by a single controller synthesis step, where we find a controller that simultaneously stabilizes all the samples and minimizes performance at the same time. This is done using `hinfstruct` [9], [10].

C. Parameter-dependent D-scales

Let a parametric uncertainty sample $\Delta_p \in \Delta_p$ and controller $K(\kappa)$ be given. The corresponding closed-loop $T(\Delta_p, \kappa)$, shown in Fig. 2, is defined as

$$T(\Delta_p, \kappa) := \mathcal{F}_L(\mathcal{F}_U(M, \Delta_p), K(\kappa)). \quad (8)$$

The worst-case gain for the closed-loop over the remaining dynamic uncertainty is given by

$$\max_{\Delta_d \in \Delta_d} \|\mathcal{F}_U(T(\Delta_p, \kappa), \Delta_d)\|_\infty. \quad (9)$$

It follows from standard μ analysis results [22], [21] that an upper bound on this worst-case gain over Δ_d is given by

$$\begin{aligned} & \min_{\substack{D_d \in \mathbb{D}_d \\ \gamma > 0}} \gamma \\ & \text{subject to:} \\ & \left\| \begin{bmatrix} D_d & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} T(\Delta_p, \kappa) \begin{bmatrix} D_d^{-1} & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} \right\|_\infty \leq 1. \end{aligned} \quad (10)$$

Fig. 2 shows the sampled closed-loop with D-scales and performance gain scaling. For the remainder of the derivation, the notational dependence of T on (Δ_p, κ) is dropped for simplicity. Form the partitioning $T =: \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$ where T_2 has the row dimension of the error signal e . Recall that, for any complex matrix L , it follows that $\bar{\sigma}(L) \leq 1$ if and only if $I - L^* L \succeq 0$. Here, $\succeq 0$ means positive semidefiniteness of the left-hand-side, and L^* is the conjugate transpose of L . This can be used to express the constraint in (10) as

a (frequency-dependent) LMI (Linear Matrix Inequality). By Schur complement lemma, this constraint is equivalent to

$$\begin{bmatrix} \gamma I & & T_2 \\ T_2^* & \begin{bmatrix} X & 0 \\ 0 & \gamma I \end{bmatrix} - T^* \begin{bmatrix} X & 0 \\ 0 & 0 \end{bmatrix} T \end{bmatrix} \succeq 0, \quad \forall \omega, \quad (11)$$

where $X := D_d^* D_d$.

Every $D_d \in \mathbb{D}_d$ has the structure $\text{diag}\{d_1 I, \dots, d_{N_d} I\}$, with each d_i as a stable, minimum-phase, invertible SISO (Single Input Single Output) system. Thus X has the structure $X = \text{diag}\{x_1 I, \dots, x_{N_d} I\}$, where $x_i = d_i^* d_i \geq 0$ for all ω . Express each x_i as $x_i = \psi^* \Lambda_i \psi$ where ψ is a (fixed) column vector of stable, minimum phase basis functions and $\Lambda_i = \Lambda_i^*$ are decision variables to be optimized.

The constraint in (11) is enforced on a frequency grid $\{\omega_i\}_{i=1}^{N_\omega}$. The cost function to be minimized is turned into

$$\hat{\gamma} + \sum_{i=1}^{N_\omega} \gamma_i, \quad (12)$$

where γ_i is the gain at ω_i and $\hat{\gamma}$ is the peak gain over all frequencies, i.e.

$$\hat{\gamma} \geq \gamma_i, \quad i = 1, \dots, N_\omega \quad (13)$$

is added to the constraints.

To ensure that $x_i \geq 0$ for all ω , additional KYP (Kalman-Yakubovich-Popov) lemma constraints are added [23]. This guarantees that the solution X obtained from the optimization can be spectral factorized to get a stable, minimum phase scaling D_d . Let $\psi(j\omega) = C_\psi(j\omega I - A_\psi)^{-1} B_\psi + D_\psi$, then $\psi^* \Lambda_i \psi \geq 0$ for all ω is expressed as

$$\begin{bmatrix} (j\omega I - A_\psi)^{-1} B_\psi \\ I \end{bmatrix}^* \begin{bmatrix} C_\psi^* \\ D_\psi^* \end{bmatrix} \Lambda_i \begin{bmatrix} C_\psi & D_\psi \end{bmatrix} \begin{bmatrix} (j\omega I - A_\psi)^{-1} B_\psi \\ I \end{bmatrix} \geq 0$$

for all ω . By the KYP lemma, this condition is equivalent to the existence of $Q_i = Q_i^*$ such that

$$\begin{bmatrix} A_\psi^* Q_i + Q_i A_\psi & Q_i B_\psi \\ B_\psi^* Q_i & 0 \end{bmatrix} - \begin{bmatrix} C_\psi^* \\ D_\psi^* \end{bmatrix} \Lambda_i \begin{bmatrix} C_\psi & D_\psi \end{bmatrix} \leq 0. \quad (14)$$

The optimization in (10) is reformulated as the minimization of the cost function in (12) with constraints (11), (13), and (14). This is a convex SDP (SemiDefinite Program) in the variables $\hat{\gamma}$, $\{\gamma_i\}_{i=1}^{N_\omega}$, $\{\Lambda_i\}_{i=1}^{N_d}$, and $\{Q_i\}_{i=1}^{N_d}$. Finally, this optimization is solved to compute a dynamic scaling D_d for each uncertainty sample $\Delta_p \in \Delta_{p,s}$.

D. Synthesis step

The output of the D-step is a set of dynamic scalings $\mathbb{D}_{d,s}$ each computed for a corresponding element of the parameter uncertainty set $\Delta_{p,s}$. The synthesis step optimizes the controller $K(\kappa)$ to minimize the worst-case gain over the set of scaled,

closed-loop samples. This is formulated as the following optimization:

$$\begin{aligned}
 & \min_{\substack{\kappa \in \mathbb{R}^{n_\kappa} \\ \gamma > 0}} \gamma \\
 & \text{subject to:} \\
 & \left\| \begin{bmatrix} D_d & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} T(\Delta_p, \kappa) \begin{bmatrix} D_d^{-1} & 0 \\ 0 & \frac{1}{\sqrt{\gamma}} I \end{bmatrix} \right\|_\infty \leq 1 \\
 & \forall (\Delta_p, D_d) \in \mathbb{A}_{p,s} \times \mathbb{D}_{d,s}.
 \end{aligned} \tag{15}$$

This control synthesis is directly addressable using the `systune` command in MATLAB [11].

If $K(\kappa_{\text{init}})$ is not robustly stabilizing for the dynamic uncertainty, as assumed in Algorithm 1, the D-scales are set to identity in Step 5. Under these conditions, the optimization in (15) may still yield a solution (and in fact usually does). If there is no solution however, a different algorithm is invoked, which maximizes the stability margin of the system, thus provides a robustly stabilizing controller if possible. This algorithm is not detailed in this paper for lack of space.

E. Analysis step

The analysis step computes upper and lower bounds on the worst-case gain of the closed-loop given in (6) for a fixed controller $K(\kappa)$. This is performed using the `wcgain` function in MATLAB but modified to incorporate branch-and-bound on the parametric uncertainty. The lower bound γ_a is computed using a combination of a (skewed- μ) power iteration for the dynamic uncertainty blocks and a Hamiltonian-based coordinate-wise iteration on the parametric uncertainty. This returns a worst-case uncertainty Δ^* that achieves the lower bound. The upper bound $\bar{\gamma}_a$ is computed using a frequency-gridded SDP with D-scales for the dynamic uncertainty and (D, G) -scales for the parametric uncertainty.

Branch-and-bounding of the real uncertainty is used to reduce the gap between the upper and lower bounds. Specifically, the real parameter uncertainty set \mathbb{A}_p is described by a normalized hypercube. This hypercube is split and the upper/lower bounds are computed on each sub-cube. The splitting of sub-cubes continues until the gap between the overall upper and lower bounds is below some relative error or a maximum number of cube splits is reached. Details on this analysis step are given in [24].

When the worst-case gain upper bound $\bar{\gamma}_a$ is infinite, the worst-case uncertainty is chosen which corresponds to the upper bound of the stability margin of the closed-loop. This is computed using the `robstab` function in MATLAB [25].

III. FLUTTER SUPPRESSION CONTROL DESIGN FOR THE FLEXIBLE AIRCRAFT

The demonstrator of the FLEXOP project, shown in Fig. 3, was built specifically for flutter control experimentation [16]. The single-engined aircraft features a wing span of 7 m, aspect ratio of 20, and takeoff weight between 55 and 66 kg. This section is dedicated to the application of the structured synthesis method to the flutter suppression control design for this platform. In Section III-A, the high fidelity model is given



Fig. 3. The demonstrator aircraft built for the FLEXOP project.

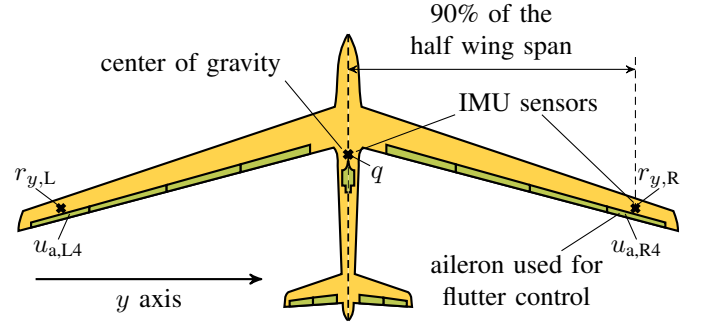


Fig. 4. Positions of the sensors and control surfaces used for the flutter suppression control. The control inputs and measurements are marked at the corresponding control surface or sensor. The signals $r_{y,L}$, $r_{y,R}$, and q denote angular rates along the y axis, $u_{a,L4}$ and $u_{a,R4}$ are aileron deflection commands for the actuators.

which is used in the analysis of the closed loop. The reduced-order control-oriented model is described in Section III-B. Finally, the formulation of the uncertain design model and the control problem along with the synthesis results are in Section III-C.

A. Nonlinear high fidelity model of the flexible aircraft

Each wing of the FLEXOP aircraft is equipped with four ailerons [26] with the outermost control surfaces dedicated to flutter suppression as illustrated in Fig. 4. In addition to the GPS and air data probe, the aircraft features inertial measurement units (IMUs) at the center of gravity and in the wings as shown in Fig. 4. A custom direct drive actuator is developed for flutter suppression in order to have adequate bandwidth for the flutter frequency. The direct drive actuator has 0.1 ms delay and has the transfer function

$$G_{\text{act}}(s) = \frac{0.78s + 2.74 \cdot 10^5}{s^2 + 564.51s + 2.74 \cdot 10^5}. \tag{16}$$

The construction of the aeroservoelastic (ASE) model is based on a subsystem approach which involves the integration of aerodynamics, structural dynamics, and flight dynamics [27], [28] as illustrated in Fig. 5. The components in Fig. 5 are developed separately and then combined to form the ASE model.

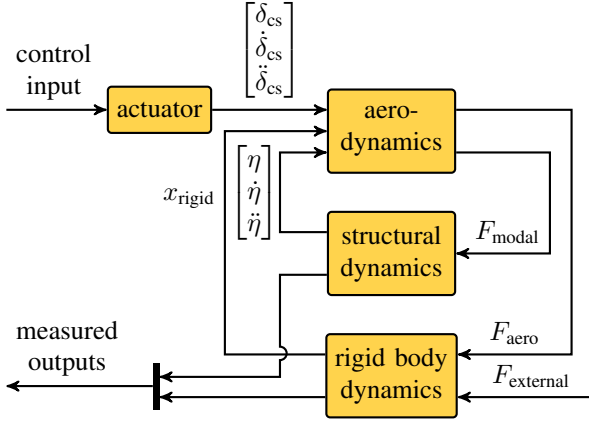


Fig. 5. ASE subsystem interconnection. (F_{aero} represents the aerodynamic forces acting on the rigid body dynamics, F_{external} represents external i.e. the propulsion and gravitational forces; the rest of the variables are defined in the remainder of the section.)

The structural dynamics are modeled by high fidelity finite element method [29] and are then condensed with Guyan reduction [30]. The aerodynamics model of the aircraft is based on the vortex-lattice method (VLM) and the doublet-lattice method (DLM) and it is complemented by results from computational fluid dynamics (CFD) simulations. The nonlinear equations of motions are derived based on a mean axes reference frame [31]. The mean axes approach describes the dynamics of the flexible body by a set of equations which decouple the rigid body modes from the vibrational modes. The mean axes coordinates ensure that the coupling is restricted to external forcing terms only [31].

The nonlinear ASE model of the FLEXOP aircraft consists of 12 rigid body states, 100 states corresponding to flexible modes, and 1040 aerodynamic lag states in addition to the actuator dynamics. This model is considered as the high fidelity model. Based on this model, the aircraft has two unstable aeroelastic modes. The symmetric flutter mode becomes unstable at 52 m/s and 50.2 rad/s and the asymmetric mode becomes unstable at 55 m/s and 45.8 rad/s.

B. Bottom-up model reduction

The high fidelity model in III-A has over one thousand states. Control design for such a high dimensional model is not possible therefore an appropriately low order and numerically tractable control-oriented model is required [17]. Because the size of this model poses considerable difficulty to the linear parameter varying reduction techniques [32], [33], the "bottom-up" modeling approach in [34], [29] is applied in this paper. The key idea is to reduce the components of the system in Fig. 5 separately. Since the structural and aerodynamics subsystems have simpler structure than the combined ASE model, it is possible to simplify them using more straight forward reduction techniques. This approach leads to a sufficiently low order ASE control-oriented model.

The control surfaces used for flutter suppression are the two outermost ailerons of the aircraft as illustrated in Fig. 4. The actuating signal is the control surface deflection command

received by the actuator denoted as $u_{a,L4}$ and $u_{a,R4}$ for the left and right wing respectively. The sensors used for flutter control are three IMU's: one in the center of gravity and two at the 90% of the half wing span on each wing. Only the angular rate measurements along the y axis of the three IMU's are used (see Fig. 4). The signals of the IMU's on the left and right wings are denoted by $r_{y,L}$ and $r_{y,R}$ respectively while q denotes the pitch rate in the center of gravity. Only these inputs and outputs are considered in the reduction.

The frequency range of interest where the control-oriented model is expected to provide a good approximation of the high fidelity model is $[0, 100]$ rad/s. The 100 rad/s is roughly twice the flutter frequencies 50.2 and 45.8 rad/s which ensures the accurate representation of the flutter behavior. It is possible to retain the accuracy of the control-oriented model over a wider frequency range at the expense of additional dynamics.

The objective of the reduction is to decrease the number of states of the subsystems in Fig. 5 while maintaining and acceptably low ν -gap between the high fidelity and the control-oriented model in the frequency range of interest ($[0, 100]$ rad/s). The ν -gap metric $\delta_\nu(\cdot, \cdot)$ is used since it takes into account the feedback control objective. It assumes values between zero (for identical systems) and one. If a feedback controller stabilizes the system $P_1(s)$ with stability margin ε , then it also stabilizes $P_2(s)$ if $\delta_\nu(P_1(s), P_2(s)) < \varepsilon$. A plant at a distance greater than ε from P_1 , on the other hand, will in general not be stabilized by the same controller [35]. The ν -gap between $P_1(s)$ and $P_2(s)$ at the fixed frequency ω is

$$\delta_\nu(P_1(j\omega), P_2(j\omega)) = \bar{\sigma} \left[(I + P_2(j\omega) P_2^*(j\omega))^{-1/2} \cdot (P_1(j\omega) - P_2(j\omega)) (I + P_1^*(j\omega) P_1(j\omega))^{-1/2} \right]_2, \quad (17)$$

where $\bar{\sigma}[\cdot]$ denotes largest singular value.

The aerodynamic lag terms assume the state-space form

$$\begin{aligned} \dot{x}_{\text{aero}} &= \frac{2V_{\text{TAS}}}{\bar{c}} A_{\text{lag}} x_{\text{aero}} + B_{\text{lag}} \begin{bmatrix} \dot{x}_{\text{rigid}} \\ \dot{\eta} \\ \dot{\delta}_{\text{cs}} \end{bmatrix} \\ y_{\text{aero}} &= C_{\text{lag}} x_{\text{aero}} \end{aligned} \quad (18)$$

where V_{TAS} is the true airspeed, \bar{c} is the reference chord, x_{rigid} is the rigid body state, η is the state of the structural dynamics, and δ_{cs} is the control surface deflection.

Balancing transformation is applied for the aerodynamics model given by A_{lag} , B_{lag} and C_{lag} in (18). The order reduction is achieved by residualizing the states with the smallest Hankel singular values. Keeping two lag states results in acceptable accuracy. Coefficient $C_{Q_{3\eta_1}}$ for the modal generalized force (see Chapter 7 of [31] for more details) affecting the symmetric flutter mode and coefficient C_{l_p} affecting the asymmetric flutter mode of the control-oriented model were scaled by 0.65 and 1.6 respectively. By this heuristic modification, the resulting control-oriented model matches the flutter speeds and frequencies of the high fidelity model better. The effect of this modification to the rest of the dynamics is negligible.

The structural dynamics of the aircraft are of the form

$$\mathcal{M}\ddot{\eta} + \mathcal{C}\dot{\eta} + \mathcal{K}\eta = F_{\text{modal}} \quad (19)$$

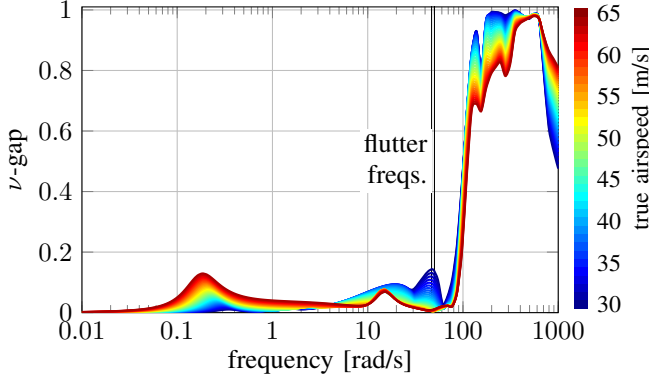


Fig. 6. The ν -gap values as a function of frequency between the high fidelity and the control-oriented model for the inputs and outputs used for the flutter suppression control design.

where \mathcal{M} , \mathcal{C} and \mathcal{K} are the modal mass, damping and stiffness matrices respectively and F_{modal} is the external excitation in modal coordinates. The structural dynamics model is an LTI system therefore state truncation is applied. Along with the first six structural modes, modes 19, 20, 21 are retained since their removal results in a large increase in the ν -gap between the high fidelity and the control-oriented model. This way, the 100th order structural dynamics model is reduced to 18 states. It is assumed that the structural dynamics model has parametric uncertainty. Specifically, the first six modes of the control-oriented model have $\pm 1\%$ uncertainty in the natural frequency and $\pm 10\%$ in their damping.

The resulting bottom-up control-oriented model has 56 states that consists of 12 rigid body states, 18 structural dynamics states, 2 aerodynamic lag states, and 24 actuator dynamics states. The design model is obtained by trimming and linearizing this model for straight and level flight at 36 equidistant points of the airspeed in the interval $[30, 65]$ m/s and for each combination of the perturbed parameters in the structural dynamics. The ν -gap between the high fidelity and the control-oriented model for different airspeed values and for the nominal structural dynamics is depicted in Fig. 6. The ν -gap is calculated for the inputs and outputs used for flutter control (see Fig. 4).

The pole migration of the high fidelity and the control-oriented model is compared in Fig. 7. The full order model predicts flutter at 52 and 55 m/s at frequencies of 50.2 and 45.8 rad/s respectively. In comparison, flutter occurs in the control-oriented model at 52 and 56.5 m/s at 50.3 and 46 rad/s respectively. The flutter speed and frequency accuracy of the control-oriented model is deemed sufficient for control design.

C. Flutter suppression control synthesis

In this Section, the flutter suppression control design for the flexible aircraft in Section III-A is discussed. First, two uncertain SISO models are obtained from the reduced order control-oriented model detailed in Section III-B. Then, the performance specification used for the optimal control design

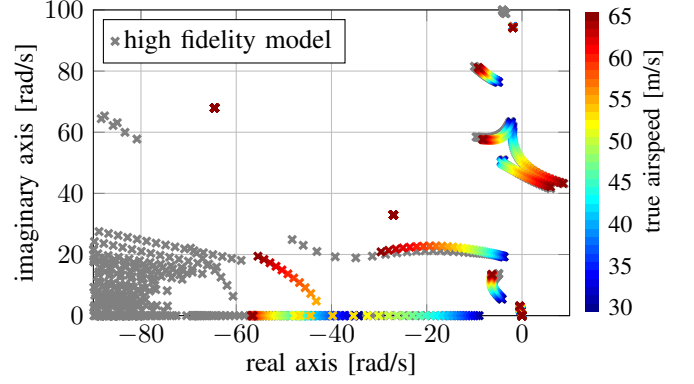


Fig. 7. Pole migration of the control-oriented and high-fidelity models.

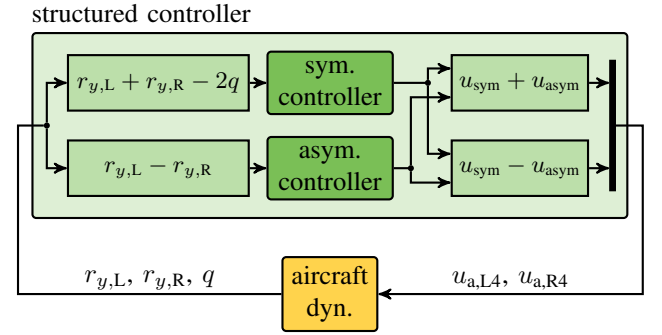


Fig. 8. Structure of the control loop using two SISO controllers to stabilize the symmetric and asymmetric flutter modes separately.

is described. Finally the results of the synthesis method in Section II are given.

1) *Design model* : Selecting the control surfaces and sensor signals as described in Section III-B (and in Fig. 4) allows us to separate the symmetric and asymmetric flutter modes of the aircraft using the combination of these variables illustrated in Fig. 8. Two SISO systems are obtained this way: $\tilde{G}_{\text{sym}}(s)$ and $\tilde{G}_{\text{asym}}(s)$. The input and output of $\tilde{G}_{\text{sym}}(s)$ are $u_{a,L4} + u_{a,R4}$ and $r_{y,L} + r_{y,R} - 2q$ respectively. The states consist of w (vertical velocity), q (pitch rate), the modal coordinates that correspond to the symmetric deformations, the lag states, and the actuator states. For $\tilde{G}_{\text{asym}}(s)$, the input and output are $u_{a,L4} - u_{a,R4}$ and $r_{y,L} - r_{y,R}$ respectively. The states are v (horizontal velocity), p (roll rate), r (yaw rate), the modal coordinates corresponding to the asymmetric deformations, and lag and actuator states.

Both parametric and dynamic uncertainty are introduced using $\tilde{G}_{\text{sym}}(s)$ and $\tilde{G}_{\text{asym}}(s)$. As the result of the model reduction technique in Section III-B, the state-space matrices of these systems are given on a parameter grid. The grid consists of 36 equidistant points of the true airspeed between 30 m/s and 65 m/s, 5 points of the natural frequency in the structural dynamics between $\pm 1\%$ of the nominal value, and 5 points of the damping in the structural dynamics between $\pm 10\%$ of the nominal value. These two arrays of LTI systems are used to introduce parametric uncertainty in the system. The dependence of the dynamics on the airspeed is expressed via the uncertain parameter δ_V . The uncertainty of the nat-

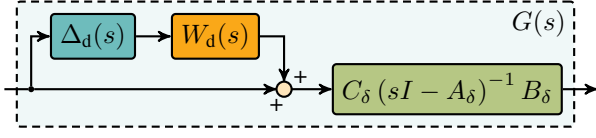


Fig. 9. Uncertain plant interconnection with the dynamic uncertainty $\Delta_d(s)$ and A_δ , B_δ , C_δ matrices depending on the parametric uncertainty.

ural frequency and the damping in the structural dynamics correspond to uncertain parameters δ_{ω_0} and δ_ξ respectively. The parameters are normalized uncertainties i.e. $|\delta_V| \leq 1$, $|\delta_{\omega_0}| \leq 1$, $|\delta_\xi| \leq 1$. Least squares fitting is performed to obtain the uncertain state-space matrices of 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 $\{A_i\}_{i=0}^4$, $\{B_i\}_{i=0}^2$, and $\{C_i\}_{i=0}^2$ are constant matrices. The elements of A_δ , B_δ , and C_δ are assumed to be a second order polynomial in δ_V based on Chapter 5 in [36]. Only A_δ depends on δ_{ω_0} and δ_ξ since the perturbation in the damping and natural frequency influences the position of the poles. Also, this form of A_δ , B_δ , and C_δ provides low error when compared to \tilde{G}_{asym} and \tilde{G}_{sym} in the parameter grid points.

Dynamic uncertainty is added to account for the model reduction in Section III-B. As illustrated in Fig. 9, input multiplicative uncertainty structure is chosen, i.e. both uncertain SISO plants have the form

$$G(s) = C_\delta (sI - A_\delta)^{-1} B_\delta (1 + W_d(s) \Delta_d(s)). \quad (20)$$

Here, $\Delta_d(s)$ is the stable SISO uncertainty block for which $\|\Delta_d(s)\|_\infty \leq 1$, $W_d(s)$ is the weight of the uncertainty, and the subscripts 'sym' and 'asym' are omitted. The weight is obtained by comparing the singular values of the system $C_\delta (sI - A_\delta)^{-1} B_\delta$ to the high fidelity model which results in

$$\begin{aligned} W_{d,\text{sym}}(s) &= \frac{33.31(s + 111.7)(s^2 + 49.17s + 2195)}{(s + 409.5)(s^2 + 97.17s + 198900)} \\ W_{d,\text{asym}}(s) &= \frac{55.703(s + 100)(s + 40)^2}{(s + 400)(s^2 + 350s + 25000)}. \end{aligned}$$

The Bode magnitude plot of both of the weights are depicted in Fig. 10. These weighting functions serve to add moderate uncertainty at low frequencies and significant uncertainty at high frequencies. The level of uncertainty starts to rise considerably at the flutter frequencies and it is above one (100%) after 100 rad/s. This is in accordance with the ν -gap results of the model reduction in Section III-B (see Fig. 6).

The resulting uncertain systems are written in LFT form as

$$G(s) = \mathcal{F}_U(N(s), \Delta(s)), \quad (21)$$

where $\Delta(s)$ is the structured uncertainty block containing the three uncertain parameters δ_V , δ_{ω_0} , and δ_ξ and the dynamic uncertainty block $\Delta_d(s)$. Therefore, for $G_{\text{sym}}(s)$ and $G_{\text{asym}}(s)$

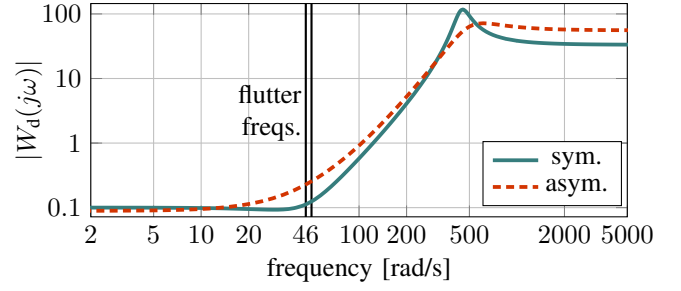


Fig. 10. Bode magnitude plot of the dynamic uncertainty weights.

respectively, the uncertainty sets Δ_{sym} and Δ_{asym} are defined as in (5) with

$$\begin{aligned} \Delta_{p,\text{sym}} &= \{\text{diag}\{\delta_V I_{51}, \delta_{\omega_0} I_{45}, \delta_\xi I_{7}\}\} \\ \Delta_{p,\text{asym}} &= \{\text{diag}\{\delta_V I_{49}, \delta_{\omega_0} I_{45}, \delta_\xi I_{7}\}\} \\ \Delta_{d,\text{sym}} &= \{\Delta_{d,\text{sym}}\} \\ \Delta_{d,\text{asym}} &= \{\Delta_{d,\text{asym}}\}. \end{aligned}$$

The number of repetitions are especially high for δ_V and δ_{ω_0} . The LFR-toolbox for MATLAB could be used to reduce the size of $\Delta(s)$ [37] but since the control synthesis method in Section II samples the uncertain parameters, this is unnecessary.

The poles of $G_{\text{sym}}(s)$ and $G_{\text{asym}}(s)$ migrate with the change of the uncertain parameters. The domain of migration of the flutter modes is illustrated in Fig. 11. The introduction of δ_{ω_0} and δ_ξ in $G_{\text{sym}}(s)$ and $G_{\text{asym}}(s)$ captures the uncertainty in frequency and damping ratio in addition to the variation with airspeed. Note that the nominal systems are stable. The gain of the two systems change due to both types of uncertainty. In Fig. 12, the Bode magnitude plot of the nominal systems are depicted along with random samples. In the frequency range below the flutter frequencies, only a moderate variation is observable while for frequencies above 100 rad/s, the gain of the systems is highly uncertain. Since the nominal systems are stable, the gain at the flutter frequencies are finite for both $G_{\text{sym}}(s)$ and $G_{\text{asym}}(s)$. For some values of the uncertainty, the gain of these systems is significantly higher or even unbounded.

2) *Performance specification* : The purpose of the flutter controller is to robustly stabilize the undamped vibrations of the aircraft. There are further considerations to be taken into account. The effect of the flutter controller on the rigid body behavior of the aircraft ought to be minimal so that the flutter controller does not interfere with the baseline controller governing the rigid body motion of the aircraft. The controller is to be implemented on an embedded computer whose sampling frequency is 200 Hz. Beyond this constraint, the bandwidth of the flutter controller must not breach the limitation posed by the actuator. Also, the closed loop must remain stable even in the presence of 15 ms output delay introduced by the autopilot hardware.

The generalized plant interconnection in Fig. 13 incorporates all the requirements above for both $G_{\text{sym}}(s)$ and

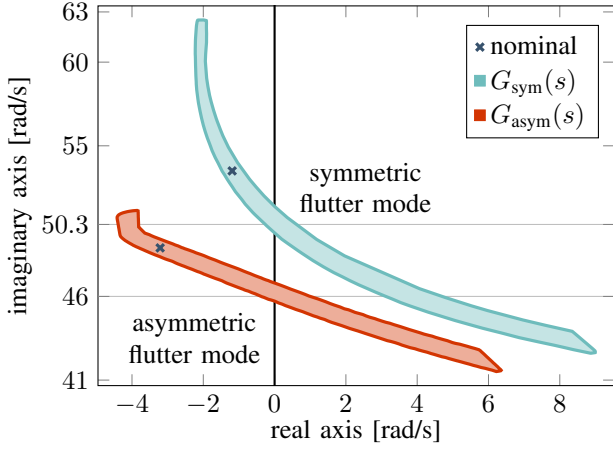


Fig. 11. Domain of the migration of the flutter mode poles due to the parametric uncertainty.

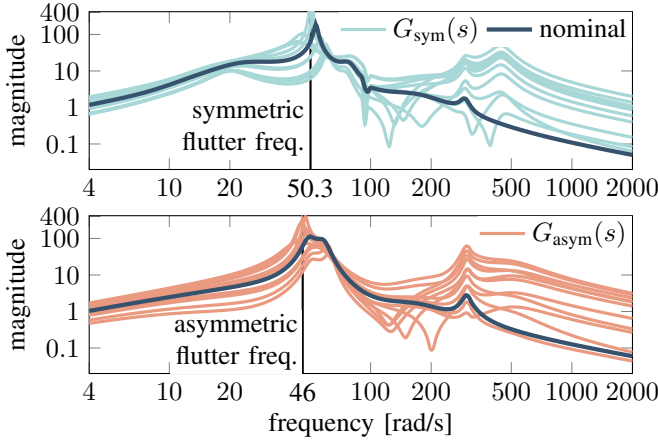


Fig. 12. Bode magnitude plot of the nominal value and random samples of the uncertain systems $G_{\text{sym}}(s)$ and $G_{\text{asym}}(s)$.

$G_{\text{asym}}(s)$. The 15 ms delay is represented by $D(s)$ that is the 4th order Padé approximation of $e^{0.015s}$. The controller is augmented with the filter

$$F(s) = \frac{1.6 \cdot 10^5}{s^2 + 560s + 1.6 \cdot 10^5} \quad (22)$$

to ensure appropriate bandwidth. This way, the sampling constraint is met and the excitation of high frequency dynamics is avoided. The Bode magnitude plot of $F(s)$ along with the performance constraints are depicted in Fig. 14.

The task of robust stabilization is expressed as sensitivity minimization. The sensitivity function of both closed loops is $S(s) = \frac{1}{1 + D(s)G(s)K(s)F(s)}$. For any stable SISO loop the minimal distance between the open loop Nyquist curve and the -1 point is the inverse of the peak of $|S(j\omega)|$, i.e. $\frac{1}{\max |S(j\omega)|} = \frac{1}{\|S(s)\|_\infty}$ [1]. It is also known that the gain margin of the loop is at least $\frac{\|S(s)\|_\infty}{\|S(s)\|_\infty - 1}$ and the phase margin is at least $2\sin^{-1}\left(\frac{1}{2\|S(s)\|_\infty}\right)$ [1]. Therefore, we want to achieve $\|S(s)\|_\infty \leq 2$ since that provides at least 6 dB gain margin and close to 30° phase margin. To that end, the weight of the

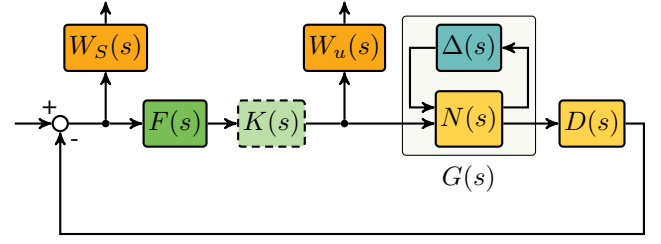


Fig. 13. Generalized plant interconnection.

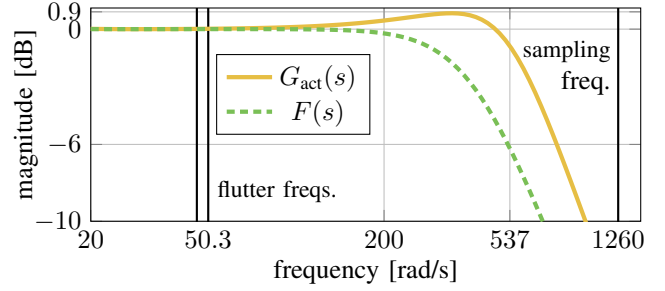


Fig. 14. Bode magnitude plot of $F(s)$ and the performance constraints. The actuator dynamics $G_{\text{act}}(s)$ are given in (16).

sensitivity is chosen as $W_s(s) = \frac{1}{s}$. To limit actuator effort, the weight of the control input is $W_u(s) = \frac{1}{10^5} = 5.78$.

3) *Synthesis* : The structure of the flutter suppression controller is depicted in Fig. 15. The SISO controllers are parametrized as general second and fourth order transfer functions, i.e.

$$K_{\text{asym}}(s) = \frac{\kappa_1 s^2 + \kappa_2 s + \kappa_3}{s^2 + \kappa_4 s + \kappa_5} \quad (23)$$

$$K_{\text{sym}}(s) = \frac{\kappa_6 s^4 + \kappa_7 s^3 + \kappa_8 s^2 + \kappa_9 s + \kappa_{10}}{s^4 + \kappa_{11} s^3 + \kappa_{12} s^2 + \kappa_{13} s + \kappa_{14}}, \quad (24)$$

where $\{\kappa_k\}_{k=1}^{14}$ are tunable design parameters. For the synthesis of $K_{\text{asym}}(s)$, a frequency grid of 30 points and 5 basis functions were used. For $K_{\text{sym}}(s)$, the number of frequency points and basis functions is 90 and 9 respectively.

As depicted Fig. 15, the flutter controller is

$$\begin{bmatrix} u_{a,L} \\ u_{a,R} \end{bmatrix} = K_f(s) \begin{bmatrix} r_{y,L} \\ r_{y,R} \\ q \end{bmatrix}, \quad (25)$$

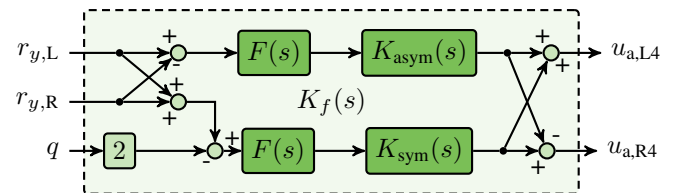


Fig. 15. Structure of the flutter suppression control.

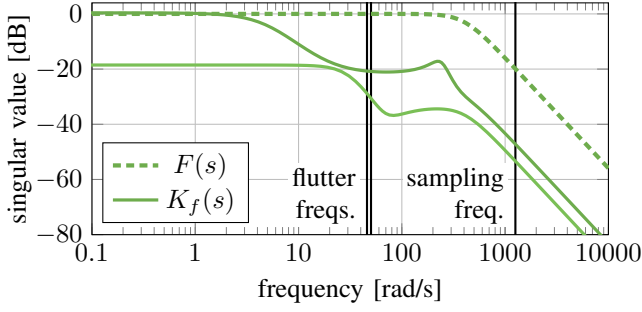


Fig. 16. Singular values of the flutter controller along with the frequency domain constraints.

where

$$K_f(s) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}. \quad (26)$$

$$\begin{bmatrix} K_{\text{asym}}(s) F(s) & 0 \\ 0 & K_{\text{sym}}(s) F(s) \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & -2 \end{bmatrix}. \quad (27)$$

Therefore, the final state order of $K_f(s)$ is 10.

The two SISO loops are tuned separately. As the result of the synthesis, the tunable components of $K_f(s)$ are

$$K_{\text{asym}}(s) = \frac{0.0106s^2 + 0.5476s + 53.6039}{s^2 + 40.4171s + 904.595}$$

$$K_{\text{sym}}(s) = \frac{0.0126s^4 + 3.7765s^3 + 1479.85s^2 + 48443.9s + 444523}{s^4 + 114.354s^3 + 59809.2s^2 + 538987s + 1.4737 \cdot 10^6}.$$

It takes 10 iterations (i.e. 10 samples of $\Delta_{p,\text{asym}}$) to obtain $K_{\text{asym}}(s)$ and 9 iterations (9 samples of $\Delta_{p,\text{sym}}$) are required for $K_{\text{sym}}(s)^\dagger$. The singular values of the flutter controller are depicted in Fig. 16. The controller has enough control authority at the flutter frequencies and it is sufficiently rolled off at the sampling frequency. The performance of $K_f(s)$ is analyzed in closed loop with the high fidelity model linearized at certain speed values.

IV. EVALUATION OF THE FLUTTER CONTROLLER USING THE NONLINEAR HIGH FIDELITY MODEL

This section details the evaluation of the flutter controller described in Section III. First, the baseline control architecture is presented that governs the rigid body motion of the aircraft. Then, the stability, performance, and robustness analysis is described with the baseline and flutter suppression control loops closed. Finally, two time domain simulations are conducted to confirm the frequency domain results. All tests are performed using the linearized versions of the nonlinear high fidelity model in Section III-A.

A. Baseline control architecture

The rigid body motion of this aircraft is described by a standard nonlinear six-degrees-of-freedom flight mechanics

[†]The computation takes approximately three hours on a computer that runs Ubuntu 16.04 LTS and features an eight-core 2.1 GHz Intel Xenon CPU with 20 GB RAM. The algorithm is run on MATLAB R2016b making use of the Parallel Computing Toolbox.

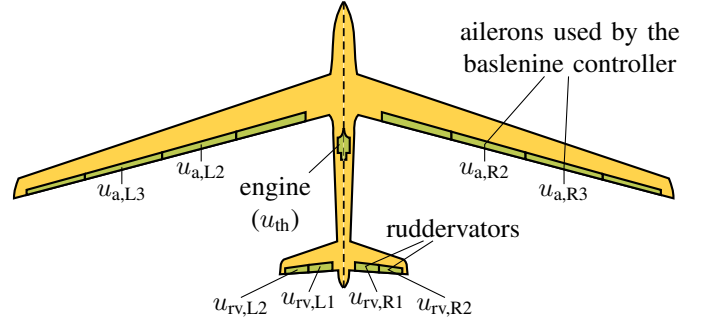


Fig. 17. Control surface configuration of the baseline control architecture. The control inputs are marked at the corresponding control surface.

model (e.g. [38]) in terms of translational velocities u , v , w and angular velocities p (roll), q (pitch), r (yaw) in the body-fixed frame. Orientation in the earth-fixed reference frame is described in terms of Euler angles Φ (bank), Θ (pitch), and Ψ (heading). The angles between the body-fixed frame and the wind axes are angle of attack α and side-slip angle β . The flight path is described with respect to the earth by the path angle γ and the course angle χ .

For the actuation of the rigid body motion, four ruddervators on the V-tail of the aircraft are used, two on the left ($u_{rv,L1}$, $u_{rv,L2}$) and two on the right side ($u_{rv,R1}$, $u_{rv,R2}$) as illustrated in Fig. 17. These ruddervators are combining the functionalities of classical rudders and elevators. The symmetric deflections of the ruddervator correspond to classical elevator deflections, while asymmetric deflections exhibit rudder deflections. Additionally, four pairs of ailerons are available. As discussed in Section III, the outermost pair ($u_{a,L4}$, $u_{a,R4}$) is used for flutter control while the innermost pair is used as high lift devices during takeoff and landing. The remaining two pairs ($u_{a,L2}$, $u_{a,R2}$, $u_{a,L3}$, $u_{a,R3}$) are utilized in the baseline control law to actuate the roll motion of the aircraft.

The structure of the baseline controller is a classical cascaded setup depicted in Fig. 18. The lateral-directional control problem is necessarily multivariable and requires the coordinated use of aileron command u_a and rudder command u_r . The innermost loop features roll-attitude (Φ) tracking, roll-damping augmentation via the roll rate (p), and coordinated turn capabilities, i.e. turns without side-slip, via feedback of the side-slip angle (β). The outer loop establishes control of the course angle (χ). All controllers are scheduled with velocity to increase performance over the velocity range. Within the fully automated flight mode the reference signals for the velocity (V_{ref}), altitude (H_{ref}), and course angle (χ_{ref}) are provided by a dedicated navigation algorithm. It uses the GPS longitudinal and lateral position of the aircraft (x_a and y_a) as well as the current course angle (χ) to provide the commands. More details on the algorithm can be found in [39].

Structure wise, the control loops use scheduled elements of proportional-integral-derivative (PID) controllers with additional roll-off filters in the inner loops to ensure that no aeroelastic mode is excited by the baseline controller. A scheduling in dependence of the true airspeed V_{TAS} is used

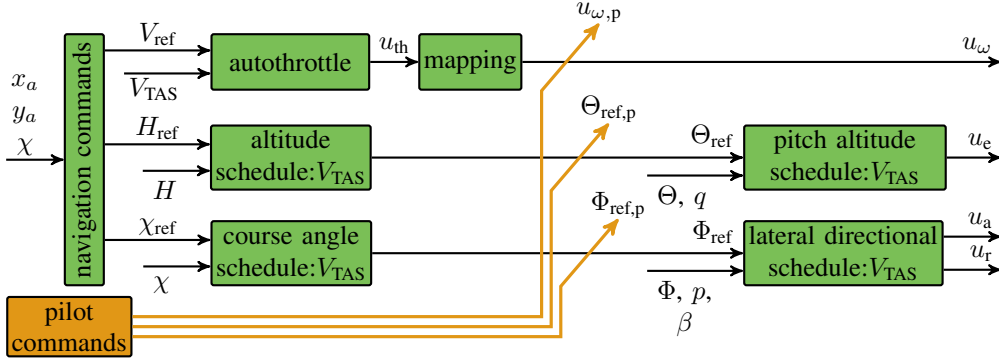


Fig. 18. Control architecture for fully automated flight, and augmented flight (indicated in orange).

TABLE I
SUMMARY OF THE CONTROL LOOPS OF THE FLEXOP BASELINE FLIGHT CONTROL SYSTEM WITH THE INNER LOOP FUNCTIONS (FIRST PART) AND AUTOPILOT FUNCTIONS (SECOND PART).

control loop	channel	structure	scheduling polyn.
pitch attitude control ($\Theta_{\text{ref}} - \Theta$) $\rightarrow u_e$		PI	2 nd -order in V_{TAS}
pitch damping $q\rightarrow u_e$		P	1 st -order in V_{TAS}
roll attitude control ($\Phi_{\text{ref}} - \Phi$) $\rightarrow u_a$		P	1 st -order in V_{TAS}
roll damping $p\rightarrow u_a$		P	1 st -order in V_{TAS}
yaw control $\beta\rightarrow u_r$		PID	2 nd -order in V_{TAS}
autothrottle $(V_{\text{ref}} - V_{\text{TAS}})\rightarrow u_{\text{th}}$		2 DOF-PID	none
altitude $(H_{\text{ref}} - H)\rightarrow \Theta_{\text{ref}}$		PI	2 nd -order in V_{TAS}
course angle $(\chi_{\text{ref}} - \chi)\rightarrow \Phi_{\text{ref}}$		PID	2 nd -order in V_{TAS}

to ensure an adequate performance over the velocity range from 32 m/s to 70 m/s. For the scheduling, a first or second order polynomial in V_{TAS} is applied. The free parameters are directly included in a structured controller optimization problem. A comprehensive summary of the used controller structures for each cascaded loop is provided in Table I, including the channel description in the controller architecture and the implemented scheduling.

Note that these controller outputs u_e , u_a , and u_r defer from the actual surface inputs to ease the control design task. Thus, they need to be transformed to physical actuator commands via an adequate control allocation. The commands to the actuators of the two aileron pairs are determined by

$$\begin{aligned} u_{a,L2} &= u_{a,L3} = 0.5u_a \\ u_{a,R2} &= u_{a,R3} = -0.5u_a \end{aligned}$$

to generate the required differential aileron deflections for roll motion control. For the ruddervators, superposition of the elevator command u_e and the rudder command u_r is applied by

$$\begin{aligned} u_{rv,L1} &= u_{rv,L2} = u_e + 0.5u_r \\ u_{rv,R1} &= u_{rv,R2} = u_e - 0.5u_r. \end{aligned}$$

Thus, symmetric deflections on the left and right of the ruddervators correspond to elevator commands while differential deflections establish rudder commands. The free parameters of the control laws are tuned to satisfy certain performance and robustness criteria as explained in [39].

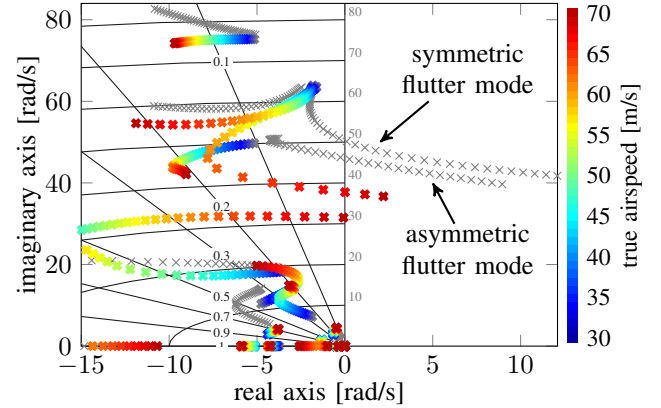


Fig. 19. Change of pole trajectories due to the flutter and baseline controller.

B. Closed-loop stability and performance evaluation

The pole trajectories of the closed loop (with both the flutter and baseline controller) are depicted in Fig. 19. The damping of both flutter modes are increased significantly. The asymmetric flutter mode is stabilized up to 70 m/s while the symmetric flutter mode crosses over to the right half plane at 68 m/s. Since the closed loop is unstable at 68 m/s, this is called the absolute flutter speed. Several other modes are influenced by the two controllers but none becomes unstable, therefore no adverse interaction between the flutter and baseline controller is observed.

Let us verify the result of the optimization in Section III in terms of the sensitivity functions. To this end, only the flutter controller is applied and the closed loop is opened loop-at-a-time to get the sensitivity function of the two SISO loops separately. The magnitudes from 30 m/s to 60 m/s are depicted in Fig. 20. The regions where the sensitivity functions are greater than one are concentrated to low frequency due to the bandwidth of the flutter controller. The sensitivity functions are also reasonably flat. The peaks are 2.41 and 2.08 for the symmetric and asymmetric loop respectively. This means, that the objective of pushing the sensitivity functions below two was not completely accomplished. The resulting stability margins are computed in Section IV-C.

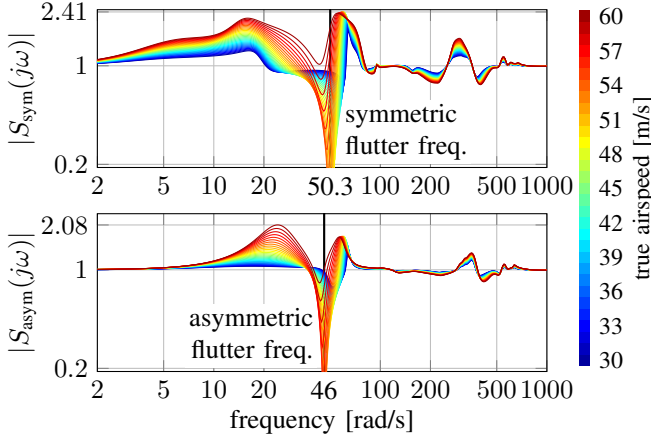


Fig. 20. Sensitivity functions of the two SISO loops.

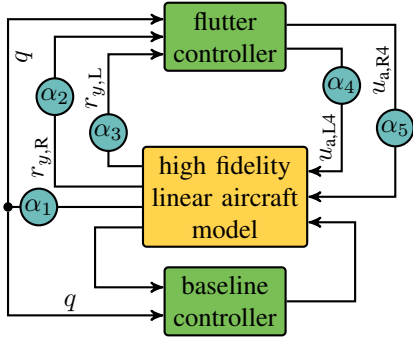


Fig. 21. Interconnection with the injected uncertainties for the disk margin analysis.

C. Robustness analysis

The robustness of the closed loop is analyzed using disk margins. To define the disk margins consider the interconnection in Fig. 21. Here, $\alpha_k \in \mathbb{C}$ are complex scalar uncertainties of the form $\alpha_k = \frac{1+\delta_k}{1-\delta_k}$ where $\delta_k \in \mathbb{C}$, $k = 1, \dots, 5$. We differentiate between loop-at-a-time and multi-loop disk margins. The loop-at-a-time disk margin is the largest simultaneous gain and phase variation in a single channel for which the closed loop remains stable. It is the largest value of m_l such that the closed loop in Fig. 21 is well posed and stable for $|\delta_l| < m_l$ while $\delta_k = 0$ for $k \neq l$. For multi-loop-margins, the uncertainties in several loops are allowed to vary simultaneously and independently.

The classical gain and phase margin interpretation of the loop-at-a-time margins for each input and output channel in Fig. 21 is depicted in Fig. 22. For four channels, the margins are nearly the same. Since the q channel is shared by the two control loops, it shows greater robustness. The loop-at-a-time margins have acceptable values up to 60 m/s where for all channels, the gain and phase margins are 5.5 dB and 34° respectively. The margins also degrade slowly after 60 m/s therefore there is a sufficiently wide safety region around 60 m/s. In view of these facts, the robust flutter speed, i.e. the speed at which the demonstrator may fly safely, is determined to be 60 m/s. The loop-at-a-time margins became zero at

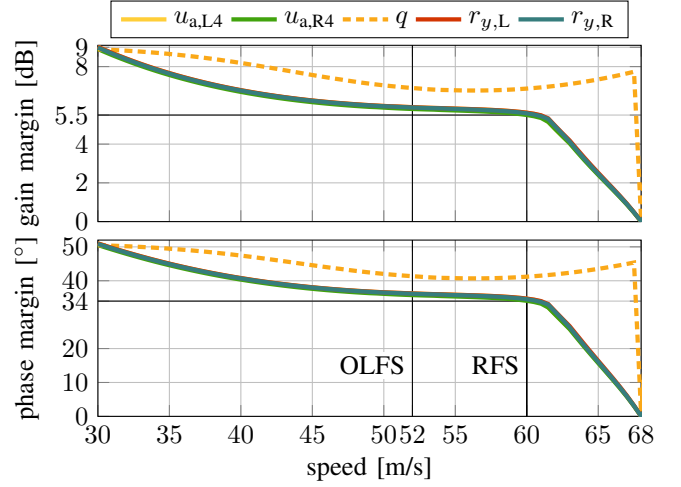


Fig. 22. Loop-at-a-time margins. (OLFS: open-loop flutter speed, RFS: robust flutter speed)

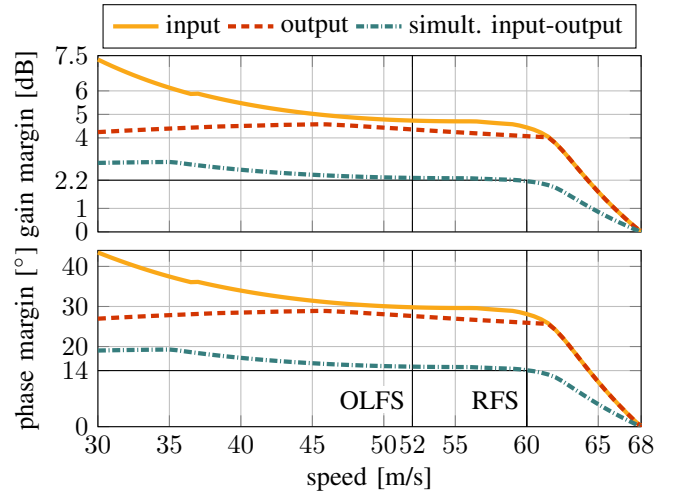


Fig. 23. Multi-loop margins. (OLFS: open-loop flutter speed, RFS: robust flutter speed)

68 m/s which is the absolute flutter speed already established in Section IV-B.

The multi-loop-margins of the closed loop are depicted in Fig. 23. These margins also start to significantly degrade after 60 m/s. The most notable is the simultaneous input/output margin for which all five uncertainties are allowed to vary at the same time. At the robust flutter speed, the simultaneous input/output margin is 2.2 dB and 14° . The rate of degradation for these margins is similar to the loop-at-a-time margins. This confirms the robust flutter speed to be 60 m/s.

The critical frequency corresponding to a loop margin is the frequency where the Nyquist curve of the open loop touches the critical point (-1 or 0). The critical frequency for all margins and for all channels are less than 65 rad/s. This ensures that these robustness measures are meaningful because it shows that the perturbation causes instability in the low frequency range where the accuracy of the modeling is

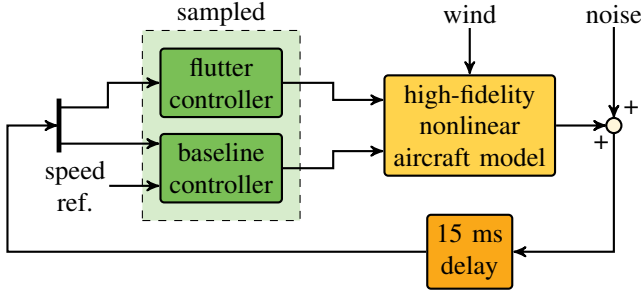


Fig. 24. Nonlinear simulation setup.

acceptable. At higher frequencies, any model is more uncertain therefore larger gain and phase margin are required.

D. Time domain simulations

Two time domain simulations are conducted to verify the frequency domain results in Sections IV-B and IV-C. In both cases, the baseline controller described in Section IV-A and the flutter controller from Section III-C3 are operating at the same time. The setup for both simulations is depicted in Fig. 24. The controllers are implemented in discrete time (with 200 Hz sampling frequency) and 15 s output delay is included to simulate the delay caused by the sensors. Gaussian measurement noise is added to the plant output. The only reference command input of the baseline controller used is the speed reference to demonstrate the behavior of the closed loop at different speed values.

In the first simulation, the speed is increased starting at 46 m/s and following a staircase reference. No wind disturbance is used in this simulation. The vertical acceleration from the two IMU's on the wing in Fig. 4 is recorded to demonstrate the load caused by the oscillations. These are denoted by $a_{z,L}$ and $a_{z,R}$ for the left and right wing respectively. The results are illustrated in Fig. 25. In the graph on the top, the airspeed is depicted as a function of time while the graph on the bottom shows the vertical acceleration. If the flutter suppression controller is not activated and the aircraft passes the open-loop flutter speed, the acceleration of the wingtip becomes greater than g indicating structural failure. With the flutter controller however, the demonstrator safely passes the open-loop flutter speed. The wingtip acceleration values are below critical at the robust flutter speed as well. The oscillations become damaging at 63 m/s, i.e. 3 m/s above the robust flutter speed. This is achieved with $|u_{a,L4}| < 0.5^\circ$ and $|u_{a,R4}| < 0.5^\circ$.

In the second simulation, we test whether it is indeed possible to fly at the robust flutter speed even in the presence of wind gusts. The aircraft is accelerated from 43 m/s to 60 m/s and then it follows a constant 60 m/s speed reference command. This is illustrated in the top diagram of Fig. 26. Without flutter control, the wing tip accelerations become critical a few seconds after the open-loop flutter speed was passed. When the flutter controller is used, the robust flutter speed is reached and acceptable acceleration values are maintained even if the wind gusts continually push the speed beyond the robust flutter speed. This demonstrates the robustness of the flutter controller

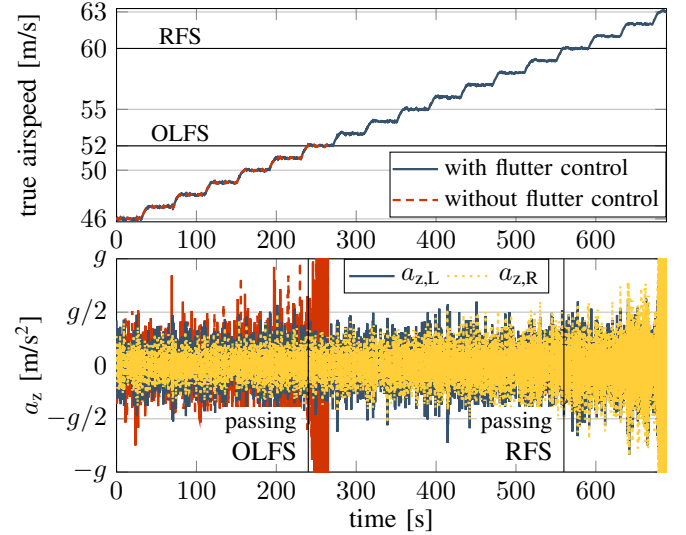


Fig. 25. Results of the first simulation. On the bottom diagram, $g = 9.81 \text{ m/s}^2$. (OLFS: open-loop flutter speed, RFS: robust flutter speed)

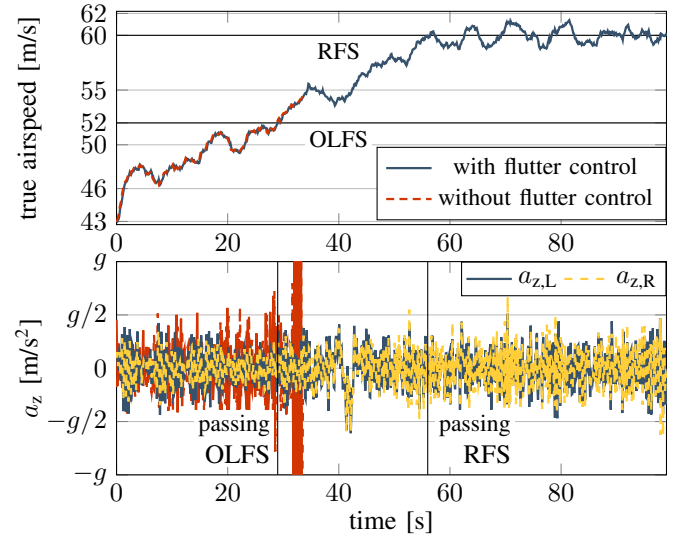


Fig. 26. Results of the second simulation. On the bottom diagram, $g = 9.81 \text{ m/s}^2$. (OLFS: open-loop flutter speed, RFS: robust flutter speed)

discussed in Section IV-C. The control input is depicted in Fig. 27. The aileron deflections increase when the open-loop and the robust flutter speed is passed. For the entire simulation, the control input is within the $\pm 1.5^\circ$ bounds.

V. CONCLUSION

A control design algorithm is presented to minimize the worst-case gain of the closed loop by tuning a fixed structure controller in the presence of mixed uncertainty. The method is iterative. In the analysis step, the worst-case sample of the uncertain parameters are computed. In the synthesis step, D-scales are constructed by solving a convex optimization problem for each sample separately to account for the dynamic

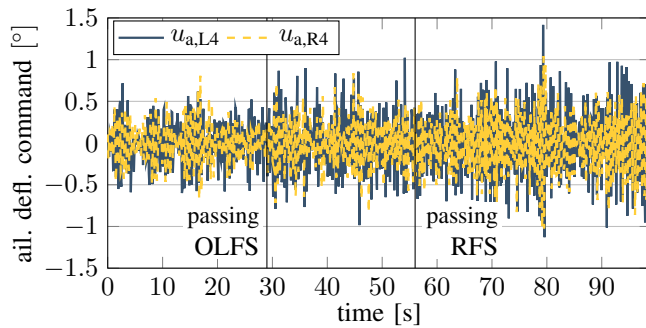


Fig. 27. Control input in the second simulation.

uncertainty. The controller is tuned for the collection of scaled samples. The applicability of this method for the flutter suppression control design of the flexible demonstrator aircraft of the FLEXOP project is also demonstrated. The problem is articulated as the minimization of the sensitivity function and control effort for two SISO loops. Disk margins are computed using the baseline controller and the high fidelity model of the aircraft. Based on these margins and on time domain simulations, the open-loop flutter speed of 52 m/s is extended to 60 m/s which is a 15% increase of the flight envelope. The controller is low order and linear time invariant therefore easy to implement on the on-board computer of the aircraft.

REFERENCES

- [1] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. John Wiley & Sons Ltd., 2005.
- [2] C. Scherer and S. Weiland, *Linear Matrix Inequalities in Control*, 2000, lecture notes.
- [3] D. Navarro-Tapia, A. Marcos, S. Bennani, and C. Roux, "Structured H_∞ control based on classical control parameters for the vega launch vehicle," in *2016 IEEE Conference on Control Applications (CCA)*, 2016, pp. 33–38.
- [4] D. Navarro-Tapia, P. Simplicio, A. Iannelli, and A. Marcos, "Robust flare control design using structured H_∞ synthesis: a civilian aircraft landing challenge," in *Proceedings of the 20th IFAC World Congress*, 9 July 2017 - 14 July 2017, ser. IFAC Papers Online, vol. 50. Elsevier Inc., 2017, pp. 3971–3976.
- [5] P. Apkarian, V. Bompard, and D. Noll, "Non-smooth structured control design with application to PID loop-shaping of a process," *International Journal of Robust and Nonlinear Control*, vol. 17, pp. 1320–1342, 2007.
- [6] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to H_∞ control," *International Journal of Robust and Nonlinear Control*, vol. 4, pp. 421–448, 1994.
- [7] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *IEEE Trans. on Automatic Control*, vol. 51, no. 2, pp. 274–286, 2006.
- [8] S. Gumussoy, D. Henrion, M. Millstone, and M. Overton, "Multiobjective robust control with HIFOO 2.0," in *IFAC Proceedings Volumes*, vol. 42, no. 6, 2009, pp. 144–149.
- [9] P. Apkarian and D. Noll, "Nonsmooth H_∞ synthesis," *IEEE Trans. on Automatic Control*, vol. 51, no. 1, pp. 71–86, 2006.
- [10] P. Gahinet and P. Apkarian, "Decentralized and fixed-structure H_∞ control in MATLAB," in *50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 8205–8210.
- [11] P. Apkarian, "Tuning controllers against multiple design requirements," in *Proceedings of the 2013 American Control Conference*, 2013, pp. 3888–3893.
- [12] B. Patartics, B. Vanek, and P. Seiler, "Structured robust synthesis with parameter-dependent D-scales," in *American Control Conference*, 2019, pp. 1800–1805.
- [13] A. Packard, J. Doyle, and G. Balas, "Linear, multivariable robust control with a μ perspective," *Transactions of the ASME*, vol. 115, pp. 426–438, 1993.
- [14] R. S. d. S. de Aguiar, P. Apkarian, and D. Noll, "Structured robust control against mixed uncertainty," *IEEE Transactions on Control Systems Technology*, 2017.
- [15] B. Patartics, P. Seiler, and B. Vanek, GitHub repository of the MATLAB implementation of the algorithm. [Online]. Available: <https://github.com/Patartics-Balint/wcgmin>
- [16] Flutter Free FLight Envelope eXpansion for ecOnomical Performance improvement (FLEXOP), *Horizon 2020 research and innovation programme of the European Union, grant agreement No 636307.*, 2015. [Online]. Available: <https://flexop.eu/>, Accessed on 11. 2019.
- [17] J. Theis, H. Pfifer, and P. Seiler, "Robust control design for active flutter suppression," in *AIAA Science and Technology Forum and Exposition*, 2016, pp. Paper No. AIAA–2016–1751.
- [18] B. Patartics, T. Luspai, T. Péni, B. Takarics, B. Vanek, and T. Kier, "Parameter varying flutter suppression control for the bah jet transport wing," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8163–8168, 2017.
- [19] M. Pusch and D. Ossmann, " H_2 -optimal blending of inputs and outputs for modal control," *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2019.
- [20] J. Doyle, "Analysis of feedback system with structured uncertainties," *IEE Proceedings D - Control Theory and Applications*, pp. 242–250, 1982.
- [21] A. Packard and J. Doyle, "The complex structured singular value," *Automatica*, vol. 29, no. 1, pp. 71–109, 1993.
- [22] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Pearson, 1995.
- [23] A. Rantzer, "On the Kalman-Yakubovich-Popov lemma," *Systems & Control Letters*, vol. 28, no. 1, pp. 7–10, 1996.
- [24] A. Packard, G. Balas, R. Liu, and J.-Y. Shin, "Results on worst-case performance assessment," in *American Control Conference*, 2000, pp. 2425–2427.
- [25] T. M. Inc., "Robust control toolbox," 2018.
- [26] C. Roessler, P. Stahl, F. Sendner, A. Hermanutz, S. Koeberle, J. Bartasevicius, V. Rozov, C. Breitsamter, M. Hornung, Y. M. Meddaikar, J. Dillinger, J. Sodja, R. D. Breuker, C. Koimtzoglou, D. Kotinis, and P. Georgopoulos, "Aircraft design and testing of FLEXOP unmanned flying demonstrator to test load alleviation and flutter suppression of high aspect ratio flexible wings," in *Proceedings of the AIAA Scitech 2019 Forum*. AIAA, jan 2019.
- [27] C. Moreno, A. Gupta, H. Pfifer, B. Taylor, and G. Balas, "Structural model identification of a small flexible aircraft," in *Proceedings of the 2014 American Control Conference*, 2014, pp. 4379–4384.
- [28] A. Kotikalpudi, "Robust flutter analysis for aeroservoelastic systems," Ph.D. dissertation, University of Minnesota, Twin Cities, 2017.
- [29] Y. M. Meddaikar, J. Dillinger, T. Klimmek, W. Krueger, M. Wuestenhagen, T. M. Kier, A. Hermanutz, M. Hornung, V. Rozov, C. Breitsamter, J. Alderman, B. Takarics, and B. Vanek, "Aircraft aeroservoelastic modelling of the FLEXOP unmanned flying demonstrator," in *Proceedings of the AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, 2019.
- [30] R. J. Guyan, "Reduction of stiffness and mass matrices," *AIAA Journal*, vol. 3, no. 2, pp. 380–380, feb 1965.
- [31] D. K. Schmidt, *Modern Flight Dynamics*. McGraw-Hill, 2012, ISBN 9780073398112.
- [32] T. Luspai, T. Péni, I. Gözse, Z. Szabó, and B. Vanek, "Model reduction for LPV systems based on approximate modal decomposition," *International Journal for Numerical Methods in Engineering*, vol. 113, no. 6, pp. 891–909, 2018.
- [33] S. Z. Rizvi, J. Mohammadpour, R. Tóth, and N. Meskin, "A kernel-based PCA approach to model reduction of linear parameter-varying systems," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1883–1891, 2016.
- [34] B. Takarics, B. Vanek, A. Kotikalpudi, and P. Seiler, "Flight control oriented bottom-up nonlinear modeling of aeroelastic vehicles," in *Proceedings of the 2018 IEEE Aerospace Conference*. IEEE, 2018.
- [35] G. Vinnicombe, "Measuring robustness of feedback systems," Ph.D. dissertation, Univ. Cambridge, Cambridge, 1993.
- [36] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.
- [37] S. Hecker, A. Varga, and J.-F. Magni, "Enhanced lfr-toolbox for matlab," *Aerospace Science and Technology*, vol. 9, no. 2, pp. 173–180, 2005.
- [38] D. T. McRuer, D. Graham, and I. Ashkenas, *Aircraft Dynamics and Automatic Control*. Princeton University Press, Princeton, NJ, 2014, vol. 740.
- [39] D. Ossmann, T. Luspai, and B. Vanek, "Baseline Flight Control System Design for an Unmanned Flutter Demonstrator," in *Proceedings of the 2019 IEEE Aerospace Conference*. IEEE, 2019.



Bálint Patartics received his M.S. degree in electrical engineering from the Budapest University of Technology and Economics, Budapest, Hungary, in 2016. He is currently an assistant research fellow at the Systems and Control Lab (SCL) of the Research Institute for Computer Science and Control (SZTAKI). His research interests include analysis of and control synthesis for uncertain systems, and flutter control.



Bálint Vaneek received his M.S. degree in mechanical engineering from the Budapest University of Technology and Economics, Budapest, Hungary, in 2003 and a Ph.D. in aerospace engineering from the University of Minnesota, Minnesota, USA in 2008. He is currently a senior research fellow and leader of the Aerospace Guidance, Navigation, and Control Group at the Systems and Control Lab (SCL) of the Research Institute for Computer Science and Control (SZTAKI). His research interests include flight control, fault detection, unmanned aerial vehicles (UAVs), inertial and satellite navigation systems.



György Lipták received his M.S. degree in computer engineering from the Budapest University of Technology and Economics, Budapest, Hungary, in 2015 and a Ph.D. degree in computer science from the University of Pannonia, Veszprém, Hungary. He was a research fellow at the Systems and Control Lab (SCL) of the Research Institute for Computer Science and Control (SZTAKI). He is currently a control engineer at Almotive. His research interests include realization and control of kinetic systems, delayed chemical networks, and reduction of linear

parameter varying systems.

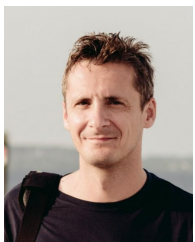


Tamás Luspai received his M.S. degree and a Ph.D. degree in traffic engineering from the Budapest University of Technology and Economics, Budapest, Hungary, in 2006 and 2011 respectively. He is currently a senior research fellow at the Systems and Control Lab (SCL) of the Research Institute for Computer Science and Control (SZTAKI). His research interests include linear parameter varying systems, analysis and reduction of large-scale dynamical systems, and modeling and control of traffic systems.



Peter Seiler received his B.S. degrees in mechanical engineering and mathematics from the University of Illinois at Urbana-Champaign, Illinois, USA, in 1996 and a Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 2001. He was a professor at the University of Minnesota in the Department of Aerospace Engineering and Mechanics. He is currently a professor at the University of Michigan in Electrical Engineering and Computer Science. His research focuses on robust control theory with applications to wind turbines and

flexible aircraft.



Béla Takarics received his M.S. and a Ph.D. degree in mechanical engineering from the Budapest University of Technology and Economics, Budapest, Hungary, in 2007 and 2012 respectively. He is currently a research fellow at the Systems and Control Lab (SCL) of the Research Institute for Computer Science and Control (SZTAKI). His research interests include multi-objective control of linear parameter varying systems via linear matrix inequality formulation, robust gain scheduling of LPV systems with integral quadratic constraints, and the modeling

and active control of aeroservoelastic vehicles.