# Flight control design for a highly flexible flutter demonstrator

Tamás Luspay*, Tamás Baár†, Dániel Teubl‡ and Bálint Vanek§
*Systems and Control Lab, Institute for Computer Science and Control, Kende u 13-17, Budapest, Hungary*

Daniel Ossmann¶, Matthias Wüstenhagen‖, Manuel Pusch** and Thiemo Kier††
*DLR German Aerospace Center, Institute of Robotics and Mechatronics, 82234 Wessling, Germany*

Sérgio Waitman‡‡, Andrea Iannelli§§, Andres Marcos¶¶ and Mark Lowenberg***
*Aerospace Engineering Department, University of Bristol, University Walk, Bristol, UK BS8 1TR*

**The paper presents the control design approaches for the European research project FLEXOP. The ultimate goal is to develop and apply active flutter suppression and load alleviation techniques on an unmanned flying aircraft demonstrator. Due to the flexible wing of the aircraft new challenges rise for the control design: the traditional rigid body (baseline) control loops have to be augmented with flutter control laws. In our approach, the controllers are designed based on a dynamical model, which is briefly discussed first. Details of the baseline control design, as well as the two different flutter suppression algorithms are discussed in the paper. Hardware-in-the-Loop testing of the controllers are reported before the first test flights of the aircraft.**

## I. Introduction

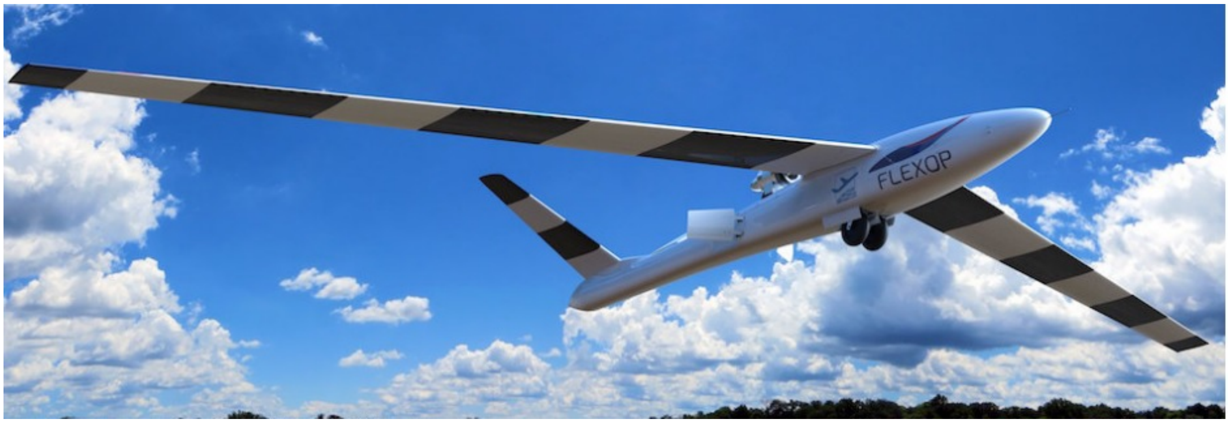


**Fig. 1 FLEXOP flutter demonstrator.**

The FLEXOP project is about developing multidisciplinary aircraft design capabilities for Europe that will increase competitiveness in terms of aircraft development costs. A closer coupling of wing aeroelasticity and flight control

*Senior Research Fellow, Systems and Control Lab
†PhD student
‡Research Engineer
§Senior Research Fellow
¶Research Engineer, DLR German Aerospace Center, Institute of Robotics and Mechatronics
‖Research Engineer
**Research Engineer
††Research Engineer
‡‡Postdoctoral Research Fellow, Aerospace Engineering Department, University of Bristol
§§Ph.D. Candidate, Aerospace Engineering Department, University of Bristol
¶¶Senior Lecturer, Aerospace Engineering Department, University of Bristol
***Professor, Aerospace Engineering Department, University of Bristol

systems in the design process opens new opportunities to explore previously inviable designs enabled by common methods and tools across the disciplines. Rapidly adapting existing designs into derivative aircraft, at a reduced technological risk (e.g. using control to solve a flutter problem discovered during development) is enabled by (a) aeroelastic tailoring to carry the redesigned derivative wing; (b) developing methods and tools for very accurate flutter modeling and flutter control synthesis, to enable improved flutter management during development, certification, and operation. Accuracy of developed tools and methods are verified on an affordable experimental platform, followed by a scale-up study, demonstrating the interdisciplinary development cycle. Manufacturers gain cost efficient methods, tools and demonstrators for enhancing aircraft performance by integrated development of flutter control and aeroelastic tailoring. These inter-disciplinary capabilities improve the design cycle and the Verification & Validation process of both derivative and new aircraft. Flight test data will be posted on the project website to provide a benchmark for the Worldwide aerospace research community. Results of the project serve as a catalyst for outlining the certification standards for future EU flexible transport aircraft.

The considered aircraft, depicted in Figure 1, is the main demonstrator of the Horizon 2020 project Flutter Free FLight Envelope eXpansion for ecOnomic Performance improvement (FLEXOP) to develop and test active flutter suppression control algorithms [1]. The single-engined demonstrator features a wing span of 7 meters. The takeoff weight is typically 55 kg but can be increased by up to 11 kg of ballast. The aircraft is equipped with a 300 N jet engine [2], located on the fuselage back. An air-brake system, deflecting from the sides of the fuselage, enables fast deceleration, fast airspeed control and steep approach angles. The empennage is configured as a V-tail, while each wing half features four control surfaces of which the outermost one is used for flutter suppression (see Figure 2). The two innermost control surfaces serve as high lift devices during takeoff and landing. In total, three pairs of wings are manufactured and will be tested on the UAV test-bench:

- Wings $-0$ – a pair of wings optimized using balanced-symmetric type of laminates serving as the reference wing, with a flutter speed far beyond the operational aircraft speed. This wing-set is mainly used for basic flight testing and rigid model verification.
- Wings $-1$ – a pair of flutter wings designed to trigger flutter within the test-regime, with two main flutter modes within the operational speed range. The flight envelope will then be extended using active flutter control.
- Wings $-2$ – a pair of wings optimized using unbalanced composite laminates, to demonstrate passive load alleviation through aeroelastic tailoring.

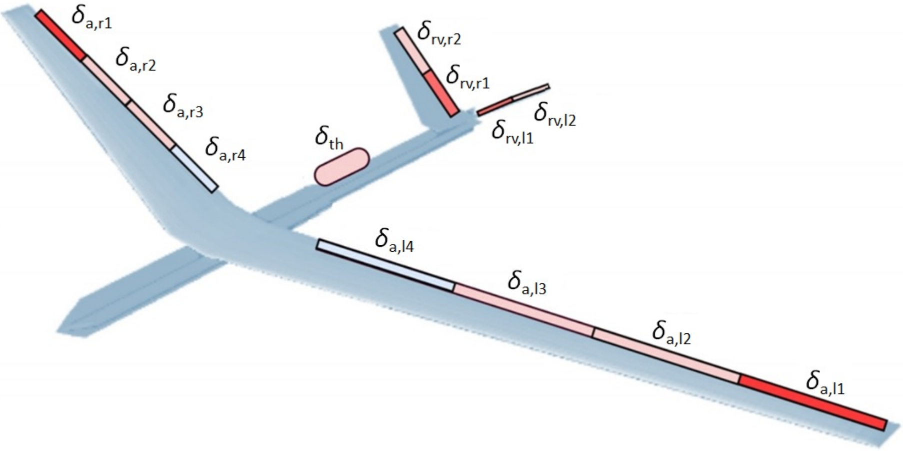


**Fig. 2 Control surface configuration.**

The paper presents the various control design efforts for the underlying flexible winged UAV test bench. The design of the aircraft is discussed in details in [3]. The control laws are derived via a model-based approach. Consequently, the first Section is devoted to the Aeroservoelastic Modeling of the aircraft. A more in-dept discussion can be found in [4]. After the modeling part, the design considerations and methodology of the baseline controller is presented, followed by two different flutter suppression control approaches. The first method - proposed by the team of DLR German Aerospace Center - applies input and output blending in order to simplify the problem for the design of individual SISO controllers. The second method - originating from the researchers at University of Bristol - is based on robust controller design and analysis techniques. Implementation and testing of the controllers are performed in a hardware in the loop environment, specifically designed for the characteristics of the FLEXOP aircraft as discussed briefly in the paper.

## II. Aeroservoelastic Modeling

This section outlines the modeling toolchain developed and applied for the FLEXOP demonstrator. The main goal of the modeling is to derive dynamical models which can serve as a basis of the controller designs as well as to give insights on the flutter phenomena.

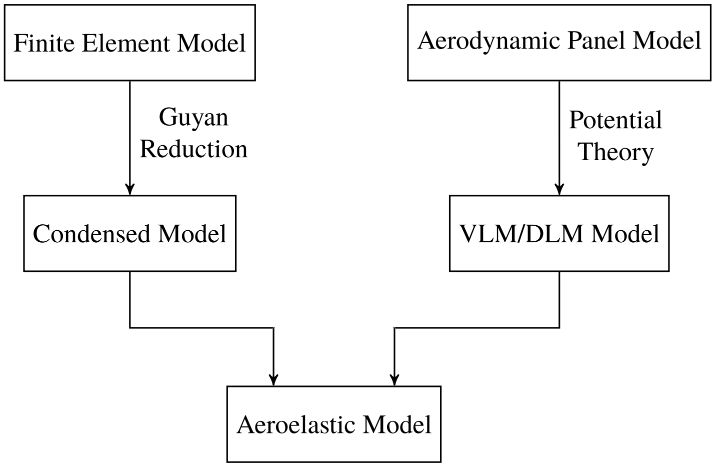


**Fig. 3 Modeling process of the demonstrator aircraft**

Figure 3 shows the modeling methodology of the underlying aircraft configuration, where it can be depicted that it involves a structural and an aerodynamic part, discussed separately in the sequel.

### A. Structural Dynamics

The structural model is based on a high fidelity finite element (FE) model, which possesses more than 600000 nodes. The number of nodes is reduced by means of the Guyan reduction, also called condensation, to less than 200 nodes. Although the number of nodes is reduced significantly the relevant dynamics with respect to flutter are still represented. A representation of the FE model and its condensation points is shown in Figure 4 [5].

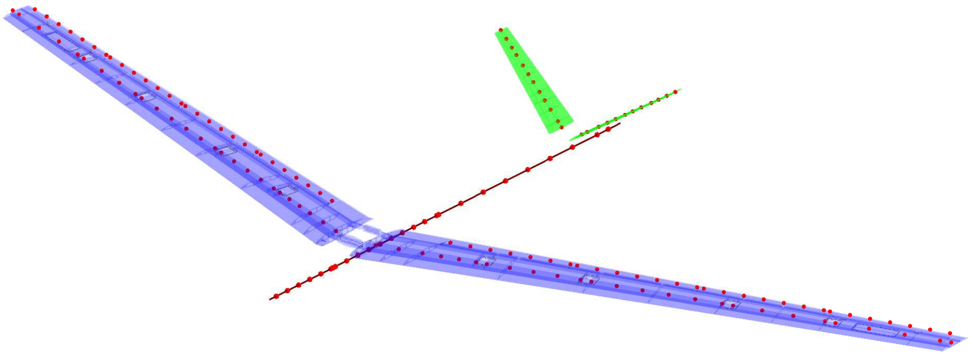


**Fig. 4 Full FE model with condensation points [5]**

The aircraft performs rigid body and flexible body motions. The non-linear rigid body motion is represented by the

Newton-Euler equations of motion [6] as:

$$\begin{bmatrix} m_b(\dot{V}_b + \Omega_b \times V_b - T_{be} g_e) \\ J_b \dot{\Omega}_b + \Omega_b \times (J_b \Omega_b) \end{bmatrix} = \Phi_{gb}^T P_g^{\text{ext}}(t). \tag{1}$$

The translational and angular velocity of the aircraft with respect to the body frame of reference are given by $V_b$ and $\Omega_b$. The vector $T_{be} g_e$ represents the gravitational acceleration, which is transformed to the body fixed frame of reference. The external loads $\Phi_{gb}^T P_g^{\text{ext}}(t)$ acting on the aircraft structure directly affect the rigid body dynamics [7].

The displacements due to the aircraft flexibility are assumed to be small, therefore linear elastic theory can be applied for the elastic motion. The correlation between external loads $P_g^{\text{ext}}(t)$ and the generalized coordinates $u_f$ representing the modal deformation of the structure is given by the differential equation:

$$M_{ff} \ddot{u}_f + B_{ff} \dot{u}_f + K_{ff} u_f = \Phi_{gf}^T P_g^{\text{ext}}(t), \tag{2}$$

where $M_{ff}$, $B_{ff}$ and $K_{ff}$ are the modal mass, damping and stiffness matrices. The modal matrix $\Phi_{gf}$ contains the eigenvectors of the structural modes sorted by frequency. It transforms the external loads into modal space [7].

### B. Aerodynamics

The aerodynamic loads mainly contribute to the external loads acting on the aircraft structure. The lifting surfaces of the aircraft are discretized by several trapezoidal shaped panels, also called aerodynamic boxes. Figure 5 depicts the aerodynamic model based on panels.

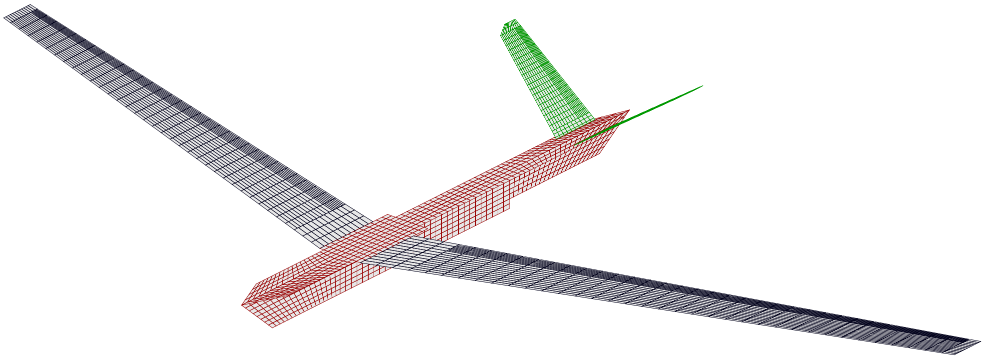


**Fig. 5 Aerodynamic boxes of the FLEXOP demonstrator aircraft [5]**

With potential theory, the vortex lattice method (VLM) for steady aerodynamics or the doublet lattice method (DLM) for unsteady aerodynamics, which are described by [5, 7], can be utilized to determine the pressure coefficients $\Delta c_{pj}$ for each panel with:

$$\Delta c_{pj} = Q_{jj} w_j. \tag{3}$$

The matrix $Q_{jj}$ comprises aerodynamic influence coefficients. It is multiplied with $w_j$, which is the down-wash $v_j$ normalized with the flight speed $U_\infty$. More information on the aerodynamic modeling of the aircraft is given by [5]. The aerodynamic forces then can be derived by

$$P_g^{\text{aero}} = q_\infty T_{kg}^T S_{kj} \Delta c_{pj}. \tag{4}$$

The integration matrix $S_{kj}$ relates the pressure in the aerodynamic boxes with the forces they cause. Through interpolation the forces of the panels are mapped onto the structural grid points with the transpose of the spline matrix $T_{kg}$. Multiplying with the dynamic pressure $q_\infty$ leads then to the aerodynamic loads [7, 8].

### C. Aeroservoelastic Model

The interaction between the aerodynamics and the structural dynamics represents the aeroelastic model. Additional components are added to the model, like shown in Figure 6. The deflection of the control surface actuators and

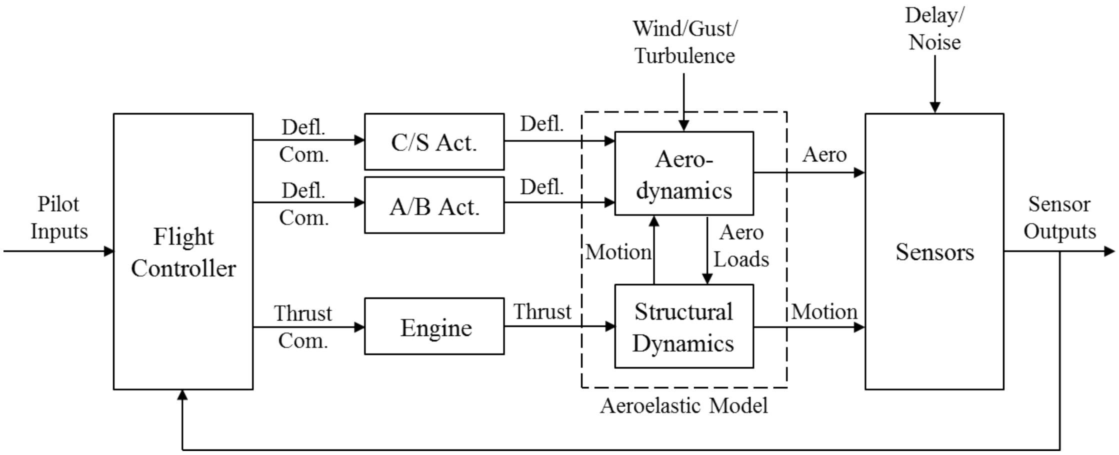


**Fig. 6 Aeroservoelastic model [5]**

airbrake actuators affect the aircraft aerodynamics, while the engine thrust acts on the structure. The aerodynamic characteristics of the aircraft are further influenced by environmental conditions like steady wind, gusts and turbulences. Readings of the onboard sensors detect the aircraft dynamics and flight conditions within the entire mission. The inertial measurement unit (IMU) sensors, installed in the wings and the fuselage, help to distinguish between rigid body motion and flexible deflection of the aircraft structure. Parameters like the barometric height, the indicated airspeed and the angle of attack are detected by airdata sensors. The position of the aircraft is provided by a GPS unit. Delay and noise are finally added to the sensor units. The sensor readings are the outputs of the system, while the pilot commands for the control surface and airbrake deflections and the thrust are the inputs. The aeroelastic representation of the aircraft dynamics is non-linear. Through linearization the dynamics can be represented by state-space systems in the form

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{5}$$

$$y(t) = Cx(t) + Du(t). \tag{6}$$

Equation 5 defines the system dynamics and correlates the effect of the inputs $u(t)$ with the states $x(t)$. Equation 6 provides the sensor dynamics, where the outputs $y(t)$ are the sensor measurements. These equations of the open-loop system are used to design control laws for the baseline and the flutter controller. When it comes to designing of the baseline controller, the system equations can be further simplified by residualization [9], as mainly the rigid body motion is of relevance. The flexible dynamics and unsteady aerodynamics can therefore be neglected, meaning the derivatives of the flexible motion states and the lag states $\dot{x}_2(t)$ are equal to zero. The state space system can then be rewritten as

$$\begin{bmatrix} \dot{x}_1(t) \\ 0 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t) = \begin{bmatrix} A_{11}x_1(t) + A_{12}x_2 + B_1u(t) \\ 0 \end{bmatrix} \tag{7}$$

$$y(t) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + Du(t) = C_1x_1(t) + C_2x_2 + Du(t), \tag{8}$$

where $x_1(t)$ are the states of which the dynamics are preserved. The second row of Equation 7 provides a definition of $x_2(t)$ with respect to $x_1(t)$:

$$x_2(t) = A_{22}^{-1}(-A_{21}x_1(t) - B_2u(t)). \tag{9}$$

This representation is favorable to derive the baseline control laws, as the system order is strongly reduced. However, for designing the flutter suppression controller different model order reduction methods have to be applied. Nevertheless, once the controllers are designed, their performance can be validated with the full order model.

## III. Baseline Control Design

This section discusses the design of the baseline controller on the basis of the reduced order model, described in the previous section. More precisely, a set of linearized systems has been obtained for various indicated airspeed values between 32 m/s to 70 m/s. The derived low order model has a state dimension of $n_x = 12$ with $n_u = 35$ inputs and $n_y = 38$ measurable outputs. Note that only a subset of these inputs and outputs are required for the baseline controller design, as the model also contains the first and second derivatives of the control surfaces along with various sensor measurements. Besides the model of the rigid body and elastic dynamics, second order actuator models, a non-linear engine model and linear sensors model are explicitly added to the design dynamics.

Three different modes of controlling the aircraft are considered in the flight control system. This facilitates a stepwise augmentation of the aircraft during the test campaigns. The three modes:

(i) **Direct Mode**: The direct mode allows the pilot to bypass the flight control system. The pilot is controlling the aircraft from a fixed position on the ground with his remote-control. The only part in the flight control computer is the mapping from the received remote-control signals to the commanded surface deflections. The pilot controls the pitch, roll and yaw axis via the surface deflections and the aircraft speed via the thrust setting.

(ii) **Augmented Mode** The augmented mode switches on basic augmentation for the pilot. Instead of directly controlling the surfaces, the pilot commands are transferred to pitch and roll attitude commands. The side slip angle is automatically commanded to zero, reducing the pilots need to control the yaw axis separately. Speed control remains in direct control, i.e. the pilot controls the speed via the thrust setting on.

(iii) **Autopilot Mode** In this mode the pilot fully delegates the aircraft control to the FCS. Altitude, course angle, speed and side slip are automatically controlled. To be able to fly along the defined test pattern, reference commands based on the aircraft position are generated in a navigation module.

### A. Control Architecture

This section provides a more detailed description of the actual control architecture of the flight control system, which is the basis for the control modes (ii) and (iii) described above. The direct mode (i) will not be further considered in the paper as a simple mapping is used to transfer the remote-control commands into the commanded surface deflections.

As control inputs the aircraft features four ruddervators, two on the left ($\delta_{rv,l1}$, $\delta_{rv,l2}$) and two on the right side ($\delta_{rv,r1}$, $\delta_{rv,r2}$) as illustrated in Figure 2. These ruddervators are combining the functionalities of classical rudders and elevators: the symmetric deflections of the ruddervator correspond to classical elevator commands, while asymmetric deflections exhibit a rudder command. Additionally, the aircraft has four pairs of ailerons. The most outer pair ($\delta_{a,l1}$, $\delta_{a,r1}$) is used for flutter control while the most inner pair ($\delta_{a,l4}$, $\delta_{a,r4}$) is used as high lift device in the approach phase. The inner two pairs ($\delta_{a,l2}$, $\delta_{a,r2}$, $\delta_{a,l3}$, $\delta_{a,r3}$) are used in the baseline control law to control the aircraft's roll motion. The engine is used for speed control.

The actuators to steer the control surfaces are modeled as second order systems with rate and position limits to realistically reflect the actuator behavior. These models have been obtained through frequency based system identification and data gathered on the various servos. The sensors of the aircraft are modeled as first order linear models including time delays. For the jet engine, a high fidelity, non-linear simulation model is available. Consequently, a simplified, control-oriented model has been developed. It features a dominant time delay of ≈ 1 s, a non-linear mapping from the engine's revolution-speed to thrust (and versa), and a rather slow second-order dynamic. In addition, a speed dependent saturation limit needed is considered, describing how the available thrust decreases with increased inflow speeds.

To facilitate the control design task and allow for the aircraft control modes (ii) and (iii), a series of cascaded control systems, illustrated in Figures 7, is used. As the cross coupling between longitudinal and lateral axis is negligible, longitudinal and lateral control design is separated. Thrust commands $\delta_{th}$ which are transferred to an engine revolution command $\delta_\omega$ via a nonlinear mapping and the elevator $\delta_e$ are the available actuators for longitudinal control. Lateral-directional control generates aileron ($\delta_a$) and rudder commands ($\delta_r$). Note that these controller outputs defer from the actual surface inputs to ease the actual control design task. It requires, however, an adequate control allocation, to generate the actual inputs commands.

The presented inputs $\delta_e$, $\delta_a$, and $\delta_r$ to the aircraft from either the flight controller or directly from the pilot are victorious inputs. This is because the aircraft has multiple surfaces for the control of each axis and also has combined rudder and elevator surface, see Figure 2. Thus, an adequate control allocation is required to generate the physical actuation commands. The aileron commands on the two pars of ailerons is determined by

$$\begin{aligned} \delta_{a,l2} &= \delta_{a,l3} = 0.5\delta_a \\ \delta_{a,r2} &= \delta_{a,r3} = -0.5\delta_a \end{aligned} \tag{10}$$

to generate the required differential aileron deflections to initiate roll motions. For the ruddervators superposition of the

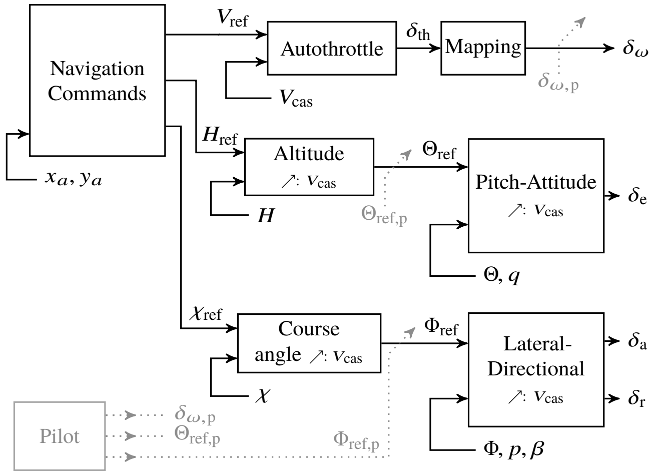


**Fig. 7 Control architecture for fully automated flight (mode (iii)), and augmented flight (mode (ii)), indicated in gray.**

elevator command $\delta_e$ and the rudder command $\delta_r$ is applied by

$$\begin{aligned} \delta_{elev,l1} &= \delta_{elev,l2} &= \delta_e + 0.5\delta_r \\ \delta_{elev,r1} &= \delta_{elev,r2} &= \delta_e - 0.5\delta_r . \end{aligned} \tag{11}$$

Thus, the symmetric deflection corresponds to the elevator command while the differential deflection corresponds to the rudder command.

The available bandwidths for throttle and elevator differ considerably such that a combined control design does not promise any advantages. The setpoint $V_{\rm ref}$ for the calibrated airspeed $V_{\rm ias}$ is controlled solely by the use of the throttle. The elevator is used to control the attitude and hence the vertical position. The pitch attitude controller in the most inner feedback loop track the pitch attitude ($\Theta$), attenuates wind disturbances, and improves short period damping with pitch rate ($q$) measurement as an auxiliary feedback signal. The cascaded outer loop establishes control of the altitude. Both controllers are scheduled with velocity ($V_{\rm ias}$), indicated by ↗ in Figure 7, to achieve optimal performance over the required velocity range. The lateral-directional control problem is necessarily multivariable and requires the coordinated use of ailerons and rudder. The most inner loop features roll attitude ($\Phi$) 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 ($\beta$). The outer loop of the cascade controllers the course angle ($\chi$). Again, both controllers are scheduled with velocity to increase performance over the speed range. Within the fully automated flight mode (iii) the reference signals for the velocity ($V_{\rm ref}$), altitude ($H_{\rm ref}$), and course angle ($\chi_{\rm ref}$) are provided by a dedicated navigation algorithm. It uses the GPS longitudinal and lateral position of the aircraft ($x$ and $y$) as well as the current aircraft velocity ($V_{\rm ias}$) to provide the commands. The algorithm is described in more detail in section 3.

In the augmented mode (ii) the pilot takes over the control of the aircraft via a remote control however, still basic augmentation is provided. Therefore, the inner loops of the control system in roll, pitch and yaw are active and receive the commands from the remote control. This is indicated in the gray in Figure 7. Not that in the direct mode (model (i)) is not depicted in the figure. In this mode the pilot uses the remote control to directly control the aileron deflections via $\delta_a$, the elevator deflections via $\delta_e$, the rudder deflections via $\delta_r$, and the engine revolution setting via $\delta_\omega$.

### B. Controller Design Approach

We have selected a scheduled, structured controller design problem [10] to address the baseline functionalities. The controller parameters in $K$ are dependent on scheduling variables described in the vector $\pi$, which is the indicated airspeed. This vector belongs to the bounded region $\Pi \in \mathcal{P}$, where $\mathcal{P}$ is the $n_p$-dimensional parameter space. The control problem is defined by

$$\begin{aligned} &\min_{K(\pi)} \max_{i,k} f_i^{(k)}(K(\pi)) \\ &\text{s.t.} \max_{j,k} g_j^{(k)}(K(\pi)) < 1 \\ &K_{\min} < K(\pi) < K_{\max}. \end{aligned} \tag{12}$$

To avoid the necessity to optimize over the multi-dimensional function space $K(\pi)$ the gains in $K(\pi)$ are restricted to polynomial basis functions of the parameters in $\pi$. For example, the $l^{th}$ element of the vector $K(\pi)$ is described by

$$k_l = a_{0,l} + a_{1,l}\pi + \cdots + a_{n_o,l}\pi^{\circ n_o}, \tag{13}$$

where $n_o$ defines the polynomial order of the basis function. The vectors $a_{m,l}$ with $m = 1, \ldots, n_o$ and $l = 1, \ldots, n_k$, are constant and have the size $1 \times n_p$. The notation $\circ$ is used to indicate that the exponent is used on each element of the parameter vector $\pi$.

For the problem considered herein the indicated airspeed is the only scheduling parameter of the controller, i.e. $\pi = V_{\text{ias}}$ and $n_p = 1$. Also, the scheduling parameter of the controller is equal to the scheduling parameter of the underlying set of linearized models. Thus, in this special case, no robustness to any unmeasurable parameter is demanded when solving the optimization problem (12).

The multi-model, multi-objective optimization problem to derive controller gains of a predefined controller structure is solved by [11]. The method converts the soft and hard constraints into normalized values and applies non-smooth optimization techniques to solve the corresponding multi-objective problem.

Depending on the range variability of the dynamics of the underlying control problem we distinguish two control design problems to be solved. In case of low variations in the dynamics properties over the aircraft velocity, the goal is to design a constant controller for the whole aircraft range. A larger change of the aircraft dynamics calls for a scheduled controller design to achieve better performance.

The soft and hard design requirements $f$ and $g$ in (12) are based on classical control objectives, including desired margins, disturbance rejection, overshoot, tracking or maximum gains and desired loop shapes. Time-domain criteria can be formalized as desired responses to *step reference signals* or *disturbance signals* in terms of response times and steady state errors. Equivalently, the desired bandwidth of the closed loop can be defined as freqeuncz domain criteria. Additionally, phase and gain *margins* can be used as criteria to improve the robustness.

Another possibility used in this paper is to provide a *reference model* and use the error between this reference model and the resulting dynamics as criteria to be minimized in (12). Such a model matching setup provides an elegant way to get as close to the desired dynamics as possible over the whole parameter range.

The presented criteria are used in this paper to optimize the controller gains $K$. More details on the actual criteria used for each cascade control loop are provided in the next section.

### C. Baseline Control Design

The rigid body motion of this aircraft is described by a standard nonlinear six-degrees-of-freedom flight mechanics model (e.g. [12]) 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 $\Phi$ (bank), $\Theta$ (pitch), and $\Psi$ (heading). The angles between body-fixed frame and wind axes are angle of attack $\alpha$ and side-slip angle $\beta$. The flight path is described with respect to earth by path angle $\gamma$, course angle $\chi$.

#### *1. Inner Loops*

The inner loops provide the basic augmentation to achieve uniform flying qualities over the whole speed range. These loops are the basis for the augmented mode (ii) to assist the pilot in flying the aircraft using the remote control. Also, the inner loops are the first cascade of the autopilot functionalities.

1) *Pitch-Attitude Control*

The pitch-attitude controller is used in the inner loop to allow the pilot to control the visible pitch attenuation of the aircraft. This is an intuitive control strategy for the pilot as the aircraft's orientation is directly controlled. A proportional-integral controller (PI) is used to ensure zero steady state error. Additionally, a pitch damper via a feedback of the pitch velocity $q$ ensures adequate damping of the short period mode of the aircraft. Omitting the time dependency for readability reasons, the controller equation is given by

$$\tilde{\delta}_e = k_\Theta^{(p)}(V_{\text{ias}})e_\Theta + k_\Theta^{(i)}(V_{\text{ias}}) \int e_\Theta dt - k_q(V_{\text{ias}})q \tag{14}$$

with $e_\Theta = \Theta_{\text{ref}} - \Theta$ with the PI controller gains $k_\Theta^{(p)}(V_{\text{ref}})$ and $k_\Theta^{(i)}(V_{\text{ref}})$ as well as the pitch damper gain $k_q^{(p)}(V_{\text{ref}})$. Note that all three gains are depended on the aircraft's velocity to ensure adequate control performance over the whole speed range. Specifically, the basis functions were selected as:

$$k(V_{ias}) = k_0 + k_1 V_{\text{ias}} + k_2 V_{\text{ias}}^2. \tag{15}$$

Adding an additional roll of filter is crucial for the design of the inners loops to ensure the frequency separation to the flutter controller as well as avoid an excitation of lightly damped modes by the baseline controller in any case. Thus the commanded signal $\tilde{\delta}_e(s)$ is filtered with a second order roll of filter with a bandwidth of 25 rad/s (a little bit less than a factor of 1.8 away from the first flutter frequency (7.2 Hz)). For the design of the pitch-attitude control gains in (14) a model matching approach is chosen. The desired tracking model is defined in terms of a second-order system with a minimum damping ration of 0.6 and corresponding frequency. Additionally a minimum phase margin of 45 degrees is defined to ensure satisfactory robustness.

2) *Lateral-Directional Control*

The lateral direction of the aircraft is controlled via the rudder and aileron deflections. The pilot is able to command the roll angle of the aircraft $\Phi$ and thereby directly control the aircraft's orientation. Additionally, the stick zero potion thereby corresponds to a wing level command, i.e. the aircraft flying straight. To provide sufficient roll damping, a roll damper is added to the controller, resulting in

$$\tilde{\delta}_a = k_\Phi^{(p)}(V_{\text{ias}})e_\Phi - k_p^{(p)}(V_{\text{ias}})p, \tag{16}$$

where $e_\Phi = \Phi_{\text{ref}} - \Phi$ is the roll angle control error, $k_\Phi^{(p)}(V_{\text{ias}})$ the proportional feedback gain for the roll angle and $k_p^{(p)}(V_{\text{ias}})$ the roll damper gain, both depending on the aircraft's velocity in a linear fashion:

$$k(V_{\text{ias}}) = k_0 + k_1 V_{\text{ias}}. \tag{17}$$

Note that through the free integral in the aircraft dynamics, i.e. $\dot{\Phi} \approx p$, a proportional feedback of the roll rate ensure also zero steady state error for roll attitude commands ($\Phi_{\text{ref}}$). This is simply due to the appearing free integral in the open loop transfer function and the resulting sensitivity reduction at low frequencies in the $\Phi_{\text{ref}} \longrightarrow \Phi$ channel. As for the longitudinal loop, an additional roll of filter is with a bandwidth of 25 rad/s is added after the control gains.

To optimize the parameter dependent controller gains, the design problem described in (12) is solved. Similarly to the pitch-attitude loop, two requirements are used. A response time of 1 s with steady-state error of 0.1 degrees is defined for the tracking behavior. Additionally, 45 deg of phase margin are defined.

To enable coordinated turn maneuver with a zero side-slip angle, the side-sleep angle $\beta$ is controlled to zero via the gain scheduled PID controller

$$\delta_r = -k_\beta^{(p)}(V_{\text{ias}})\beta - k_\beta^{(i)}(V_{\text{ias}}) \int \beta dt - k_\beta^{(d)}(V_{\text{ias}})\dot{\beta}, \tag{18}$$

where $k_\beta^{(p)}(V_{\text{ias}})$, $k_\beta^{(p)}(V_{\text{ias}})$, and $k_\beta^{(d)}(V_{\text{ias}})$ are the scheduled PID controller gains. The derivative is approximated using a first order derivative filter, with a time constant $\tau_d = 0.1$ s. The coordinated turn capabilities specially help the pilot to fly through turns much easier, as direct relationship between the orientation of the aircraft and its flight direction is ensured. In order to achieve this the gains of the side-slip loop were tuned for satisfactory disturbance rejection behavior.

### 2. *Outer Loops*

The outer loops of the control system allow to fly the aircraft fully autonomously, which is required during the flutter tests. For the flight controller in this paper the altitude and speed for the longitudinal dynamics and the flight direction for the lateral directional dynamics are chosen to be the controlled variables.

1) *Autothrottle*

The speed controller features a constant (Linear Time Invariant) two-degree-of-freedom (2-DOF) PID loop and a non-linear mapping. The feed-forward term is introduced in order to explicitly consider the low bandwidth of the jet engine and the effect of the engine's time delay for the reference command channel. With that in mind, the velocity controller equation is given by

$$\delta_{th} = k_V^{(i)} \int e_V \, dt + k_V^{(d)}(k_{ff}^{(d)} \dot{V}_{\text{ref}} - \dot{V}_{\text{ias}}) + k_V^{(p)}(V_{\text{ref}} - V_{\text{ias}}), \tag{19}$$

with the error $e_V = V_{\text{ref}} - V_{\text{ias}}$ between reference and actual speed, and the integral, derivative, and proportional gains $k_V^{(i)}$, and $k_V^{(d)}$, and $k_V^{(p)}$ respectively. The gain $k_{ff}^{(d)}$ provides an additional freedom needed to tune the feed-forward response independent of the feedback response. In other words the controller speeds up the response to reference changes without decreasing the robustness. The derivatives $\dot{V}_{\text{ref}}$ and $\dot{V}_{\text{ias}}$ of the reference speed and the feedback signal in (19) are approximated by first order derivative filters of the form $\frac{s}{\tau_d s+1}$ with a time constant $\tau_d = 0.01$ s. The presented 2DOF-PID autothrust controller provides the required thrust $\delta_{\text{th}}$, which is mapped to the required revolution-speed command $\delta_\omega$ received by the engine. Thus, the presented controller is extended by the non-linear mapping $M_{\text{th}\to\omega}$, i.e., $\delta_\omega = M_{\text{th}\to\omega}\delta_{\text{th}}$. This mapping is realized via a lookup table in the controller.

Due to the complexity of the engine and the speed feedback loop, a direct numerical optimization is selected for tuning the corresponding gains. A simulation-based approach is applied to solve the design problem and find the constant gains. In the optimization, the gains which minimize the quadratic error between the reference speed and the response of the non-linear model are determined.

2) *Altitude Control*

The outer loop for the longitudinal motion of the aircraft features an altitude reference command system. Holding the reference altitude command is crucial during the operation as any speed increase due to altitude loss needs to be avoided due to the flutter effects, which appear at high speeds.

$$\Theta_{\text{ref}} = k_H^{(p)}(V_{\text{ias}}) e_H + k_H^{(i)}(V_{\text{ias}}) \int e_H \, dt, \tag{20}$$

with $e_H = H_{\text{ref}} - H$ defining the error between the current altitude and its demand value. Here, a quadratic dependence on $V_{\text{ias}}$ is selected for the controller gains.

For the altitude tracking, a bandwidth 5 times below the resulting inner loop pitch response is desired. A factor of 5 is considered to provide a sufficient frequency separation within a cascade controller design. A quadratic dependence on $V_{\text{ias}}$ is selected for the controller gains.

3) *Course Angle Control*

To allow the tracking of way-points, a flight direction control is implemented as outer loop for the lateral control of the aircraft. Controlling the course angle $\chi$ also has the great advantage that the effect of wind effects can be compensated, and the drift of the aircraft can be reduced [13, 14]. The flight direction controller is derived a parameter depended PID controller of the form

$$\Phi_{\text{ref}} = k_\chi^{(i)}(V_{\text{ias}}) e_\chi + k_\chi^{(p)}(V_{\text{ias}}) \int e_\chi \, dt + k_\chi^{(d)}(V_{\text{ias}}) \dot{e}_\chi, \tag{21}$$

with $e_\chi = \tilde{\chi}_{\text{ref}} - \chi$. All of the controller gains are depending quadratically on the scheduling parameter $V_{\text{ias}}$. The inclusion of a derivative term in the controller structure ensures a fast (input) disturbance rejection. Again, the derivative term is replaced by a first order derivative filter. The roll angle command $\Phi_{\text{ref}}$ is additionally filtered using a second order filter of the form $\frac{16}{(s+4)^2}$ to improve the noise attenuation in the course angle feedback signal. The gains of the course angle controller in (21) are determined by demanding a response time of 5 s, which is five times slower than the roll loop. Thus, the gains are tuned in order to achieve the prescribed reference response

$$G_{\chi,\text{ref}} = \frac{1}{5s+1} \tag{22}$$

in a model matching setup within the problem setting described in (12). Again, a quadratic dependence on $V_{\text{ias}}$ is selected for the controller gains.

### *3. Navigation*

The navigation system generates the command signals for the flight control system allowing to navigate the aircraft around a predefined flight test pattern. Thus, a level above the basic flight controllers (see Figure 7) a state machine generates these required commands depending on the actual longitudinal ($x_a$) and lateral ($y_a$) aircraft positions in the earth fixed frame.

The flight test pattern is depicted in more detail in Figure 8. For the description below we are assuming without loss of generality that the north direction is equal to the y-axis of the defined coordinate system. The main waypoint to be tracked is chosen before the turn of the actual flutter test. Thus, the inbound leg is dedicated to the waypoint tracking to ensure a uniform start of the outbound turn. After the outbound turn the aircraft reaches the actual flutter test leg on the outbound leg. To test the flutter controller the aircraft speed is increased until the desired speed in the unstable region is achieved and the active flutter control techniques can be tested. On the last part of this leg the aircraft is decelerated again (activating the airbrakes if necessary), followed by the inbound turn. This results in four main sections, for which the reference signals and trajectories need to be computed. The implementation to generate the reference signals follows a state-machine with sub-tasks which are selected based on switching criteria. On the inbound leg itself, as well as during the two turns, the aircraft shall fly with a constant airspeed. The speed of is selected at $V_{\text{ref}} = V_{\text{ref},0} = 38$ m/s which is well beyond the flutter region starting at approximately 52 m/s. The waypoint tracking is a main feature of the navigation algorithm as it ensures that the aircraft show a uniform test leg each time.

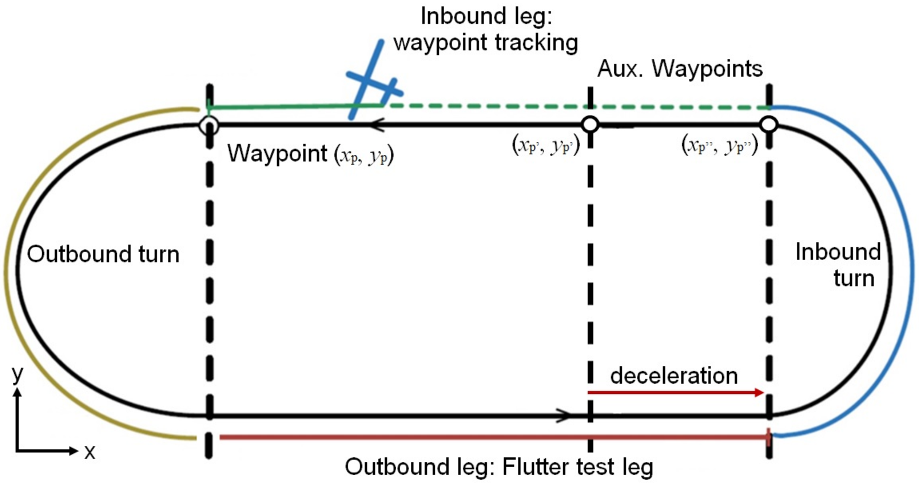


**Fig. 8 Flight Test Pattern.**

To track the defined waypoint on the inbound leg, the actual course angle reference $\chi_{\text{ref}}$ is computed based the actual position ($x_a$, $y_a$) of the aircraft. The required course angle is computed in each time step by simple geometry. The outbound turn is initiated when the aircraft reaches a horizontal distance within a 25 m radius to the way point.

The turns are flown at constant speeds at $V_{\text{ref,t}}$ in coordinated manner ($\beta = 0$), which is ensured by the baseline controller. The turns require the generation of an adequate course angle reference signal $\chi_{\text{ref,t}}$, with a constant $\Phi_{\text{ref,t}} = 40°$ desired roll angle during the turn. When the course angle reaches 30 deg to its target value the command signals is exponentially decreased to zero for a smooth transfer. The criterion for the navigation system to switch to the straight-line flights is defined by the predefined orientation of the flight test pattern. If the course angle is within 1 degree of the desired course angle the system switches the commands to the straight flight reference signals.

The outbound leg is the actual flutter test leg. Thus, on this leg the speed is increased to reach the flutter region above 52 m/s aircraft speed. This requires an adequate speed reference signal $V_{\text{ref}}$. The idea is to gradually increase the speed at each lap, according to the monitored accelerometers.

As turning the aircraft above the open loop flutter speed should be avoided, the aircraft is decelerated on the outbound leg before flying the turn. The outbound turn is initiated if the aircraft crosses the fictitious line through the second auxiliary waypoint ($x_{p''}$, $y_{p''}$) depicted in Figure 8. If the aircraft has passed this line can be determined using the same principle as presented above.

### D. Design Verification

In this section the verification results of the control systems are presented. Using the nonlinear model of the aircraft, the test pattern depicted in Figure 8 is simulated. As the during the flight test campaign the speed is increased on the test leg each lap. In the presented simulation, the initial speed is 38 m/s which is increased to 42 m/s in the first lap and 46 m/s and 50 m/s in the second and third lap.

First, the controller is validated on the described 3 lap pattern with no winds included. The resulting pattern is depicted in Figure 9. The tracks of the three laps is nearly overlapping indicating a satisfactory control performance although different speeds are flown on the test leg. Also, the waypoint to be tracked $(x_p, y_p)$, indicated by the red circle in the Figure, is tracked with an accuracy of 0.5 m in every passing.

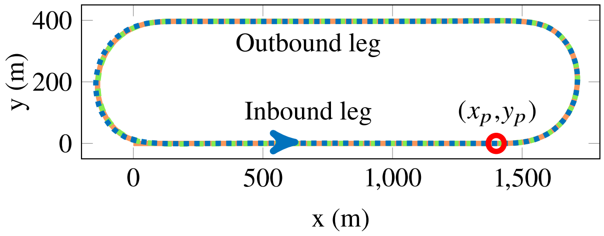


**Fig. 9 Simulated flight test pattern with three laps, lap 1 (—), lap 2 (- - -), and lap 3 (·····).**

In Figure 10 the altitude, velocity and side slip angle during the simulation is depicted. The control system is able to maintain the altitude at ±2 m around the reference altitude of 238 m. In the second diagram the indicated airspeed as well as the reference speed during the simulation is depicted. On the test leg the desired increase is achieved quickly with little overshoot. This is important to fly within the flutter region long enough to test the flutter controller but avoid large overshoots to avoid flying too deep into the instability region. Finally, the third diagram the side slip angle of the aircraft is depicted. At the beginning and the end of the turn maneuvers the side slip is slightly increased ($\approx \pm 0.2$deg), but the control system is able to bring it back to zero sufficiently fast and keep it close to zero during the turn.

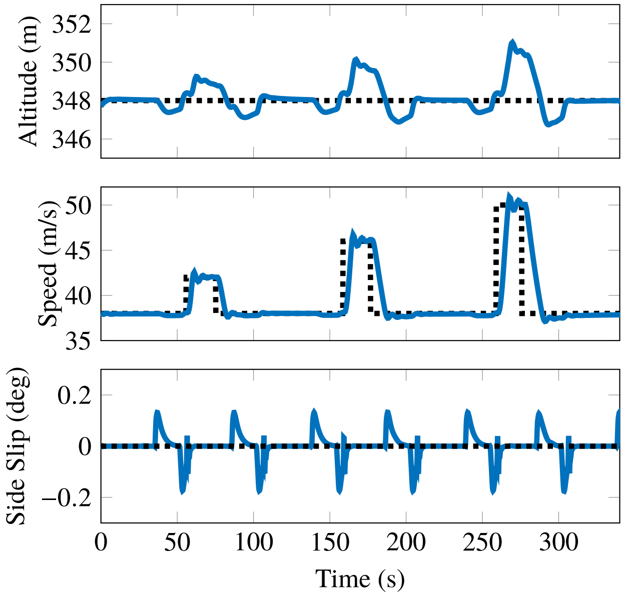


**Fig. 10 Closed loop simulation results for the altitude, velocity and side slip angle (—) in comparison to their reference signals (·····) during the three lap test pattern**

An important aspect of the simulation-based testing is, how the aircraft reacts to wind and wind gusts. First, a constant wind field with 5 m/s in the x- and y-direction is simulated. Figure 11 shows the resulting flight pattern for the same three laps as above. As expected, the whole est track is shifted into the wind direction. This is due to the fact that the actual position of the aircraft is not part of the control algorithm. What needs to be tracked, however, is the defined waypoint $(x_p, y_p)$ on the inbound leg which is achieved with accuracy of 5 m. The bigger deviations after the waypoint result from the required course angle correction before the waypoint which leads to a required turn larger than 180 degrees to get on the outbound leg. The speed tracking is quite similar to the one presented in figure 10.

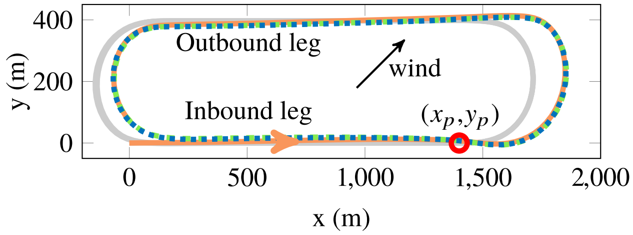


**Fig. 11 Simulated flight test pattern with constant wind for lap 1 (—), lap 2 (- - -), and lap 3 (⋯⋯) depicted together with the pattern without wind (—).**

The final scenarios include gust injections in altering directions. To get a better view on the performance the aircraft shall fly straight with a course angle of 0 deg at 348 m altitude and 38 m/s wind airspeed. Figure 12 shows the relevant flight parameters altitude, speed, course angle, and directional positions in the open and closed loop simulation. The controller keeps the aircraft in track and minimizes the variation in the mentioned flight parameters, confirming the functionality of the developed flight control system.

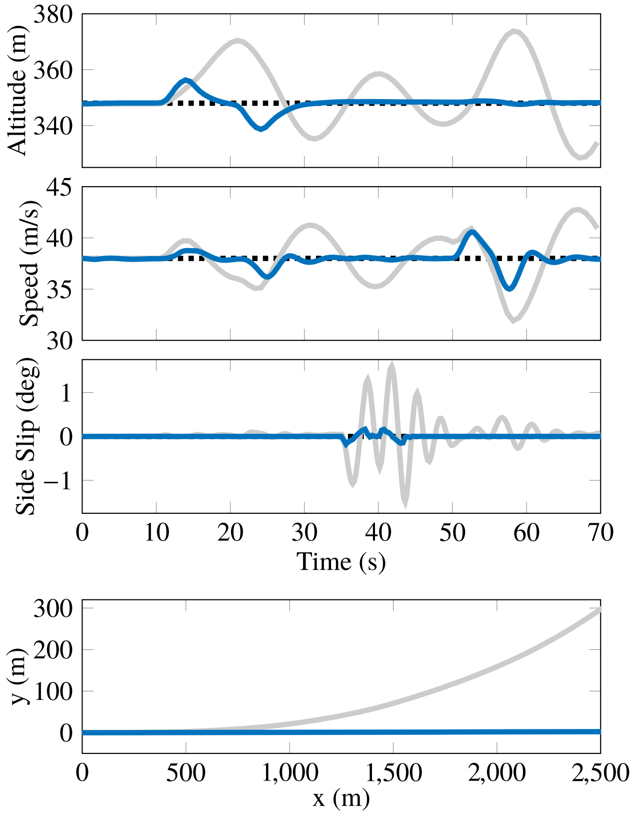


**Fig. 12 Gust simulation comparing the open (—) and closed loop (—) results to the reference signals (⋯⋯) for the controlled variables and aircraft position.**

### E. Flexible Wing-Set

The presented results so far are shown for the rigid wing-set. As the ultimate goal is to test flutter with the flexible wing-set, the next step is to test the baseline controller with the flexible wing-set. The rigid body motion,however, does not vary much with the wing-set. Thus, the provided results so fare are valid for both wing-sets. However, it is not possible to fly the flexible wing-set above 50 m/s without active flutter suppression due to the first instability occurring at 52 m/s [15]. To verify this in the non-linear simulator, the aircraft is accelerated from 38 m/s to 50 m/s. Figure 13 shows the resulting comparison of the flexible wing-set, depicted in blue, and the rigid wing-set, depicted in gray. In the first diagram the speed responses of the two simulations together with the reference signal (▪▪▪▪▪) are depicted. The results are the same as the responses are dominated by the rigid body motion for both wing-sets.

However, At 50 m/s the damping of the first bending moment decreases, which is visible, for example, in the vertical accelerations on the wing. Thus, in the second diagram of figure 13 the vertical accelerations on the left wing tip for the two wing-sets are compared. While for the rigid wing-set no oscillations appear, the flexible wing-set starts oscillating, indicating the starting instability. Simulating the aircraft model with the flexible wing-set at even higher speeds without active flutter controller does not make much sense as the highly non-linear structural and aerodynamic behavior is not covered by the model. However, we are able to show that presented baseline controller is able to bring the aircraft with the flexible wing-set to the unstable region in which active flutter control will be tested.

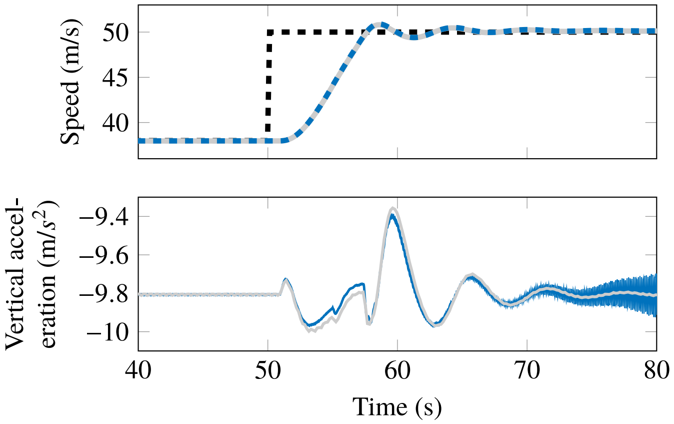


**Fig. 13 Speed response (diagram 1) and vertical wing tip acceleration (diagram 2) during an acceleration scenario from 38 m/s to 50 m/s for the flexible (blue) and the rigid (wing set).**

As follows next, two different flutter suppression controller is presented, working together with the described baseline control laws.

## IV. Aeroelastic Mode control using $\mathcal{H}_2$-optimal Blends for Inputs and Outputs

In order to increase the flutter speed of the demonstrator described above, the symmetric and anti-symmetric flutter modes need to be stabilized. Since the two flutter modes are very close in frequency, control approaches based on frequency filters are not well suited. In comparison to that, the corresponding mode shapes are orthogonal by definition, which is exploited in the modal control approach described in what follows. Taking advantage of the large number of sensors, it is proposed to weight and sum up the measurement signals such that the resulting virtual measurement output $v_{y,j}$ represents the response of the mode to be controlled. Similarly, a virtual control input $v_{u,j}$ is generated which is distributed to available control inputs such that the target mode can be individually controlled. In other words, the mode to be controlled is isolated by blending inputs and outputs. The corresponding input and output blending vectors $k_{u,j} \in \mathbb{R}^{n_u}$ and $k_{y,j} \in \mathbb{R}^{n_y}$ depend on the shape of the targeted mode and can be seen as directional filters. This implies a high robustness against frequency variations as the blending vectors are independent of the mode's natural frequency. Blending the inputs and outputs as proposed, a simple single-input and single-output (SISO) controller $c_j(s)$ can be designed to control the isolated mode.

In Figure 14, the resulting feedback interconnection is depicted, where the modes $j = 1, .., n_j$ are subject to be controlled. Summarizing the input and output blending vectors in $K_u = \begin{bmatrix} k_{u,1} & \cdots & k_{u,n_j} \end{bmatrix}$ and $K_y = \begin{bmatrix} k_{y,1} & \cdots & k_{y,n_j} \end{bmatrix}$, the

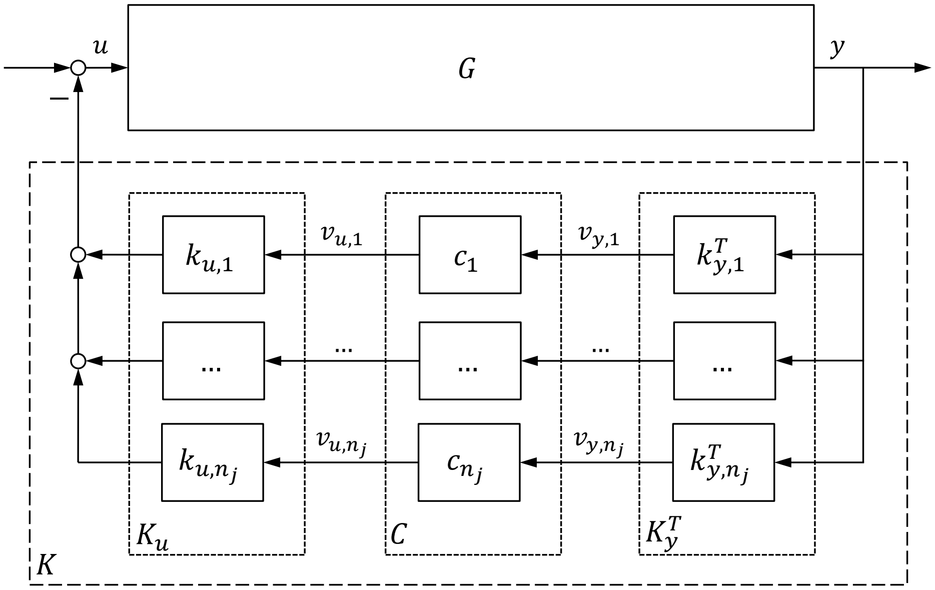


**Fig. 14 Closed-loop interconnection of the plant $G$ with the controller $K$**

overall controller is

$$K = K_u C K_y^T,$$

where the SISO controllers are collected on the diagonal of $C = \operatorname{diag}(c_1, \cdots, c_{n_j})$.

Note that the blending vector matrices $K_u$ and $K_y$ are real-valued while the individual SISO controllers are typically frequency dependent, e.g. PID controllers. Since the dynamics of the flutter demonstrator largely depend on the actual airspeed, a great improvement in controller performance is achieved when scheduling the SISO controllers with airspeed. Furthermore, also the blending vectors may be scheduled. However, simulation results show that constant blending vectors are actually sufficient to stabilize the two flutter modes within the considered flight envelope. The overall controller design procedure is schematically visualized in Figure 15, where the individual parts are described in the following subsections.

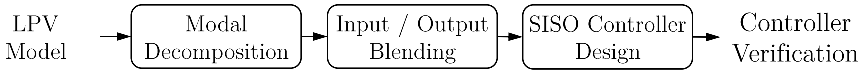


**Fig. 15 Aeroelastic mode controller design procedure.**

### A. Modal Decomposition

Linearizing the aeroservoelastic model described in [15] at a certain trim point $\rho_i$, a linear time-invariant (LTI) system is obtained which can be decomposed as

$$G_i(s) = \sum_{m=1}^{n_m} M_m(s) + D,$$

where the individual modes $m = 1, ..., n_m$ are given as

$$M_m(s) = \begin{cases} \dfrac{R_m}{s - p_m} & \text{if } \Im(p_m) = 0 \\ \dfrac{R_m}{s - p_m} + \dfrac{\overline{R}_m}{s - \overline{p}_m} & \text{otherwise.} \end{cases} \tag{23}$$

According to Equation (23), a mode $m$ is either described by a single real pole $p_m$ with an imaginary part $\Im(p_m) = 0$ or a conjugate complex pole pair $p_m$ and $\overline{p}_m$. Each pole $p_m$ is associated with a residue $R_m \in \mathbb{C}^{n_y \times n_u}$, where for a real pole, the corresponding residue is real, and for a conjugate complex pole pair the residues are also conjugate complex.

### B. $\mathcal{H}_2$-optimal Blending of Inputs and Outputs for Modal Control

The $\mathcal{H}_2$-norm of an asymptotically stable mode $M(s) = M_m(s)$ is a well-suited metric to quantify its controllability and observability in a combined way [16]. Based on this measure, $\mathcal{H}_2$-optimal input and output blending vectors $k_u$ and $k_y$ for controlling $M(s)$ are derived by solving

$$\underset{k_u \in \mathbb{R}^{n_u}, k_y \in \mathbb{R}^{n_y}}{\text{maximize}} \quad \left\| k_y^T M(s) k_u \right\|_{\mathcal{H}_2} \tag{24a}$$

$$\text{subject to} \quad \|k_u\|_2 = 1 \tag{24b}$$

$$\left\| k_y \right\|_2 = 1. \tag{24c}$$

The blending vectors $k_u$ and $k_y$, which are required to be real in order to be realizable, can be seen as input and output directions of unity length enforced by the constraints (24b) and (24c). The constrained optimization problem (24a) – (24c) has $n_u + n_y$ optimization variables and can be solved by reformulating it as an unconstrained optimization problem in a single variable, see [16] for more details.

In case the mode $M(s)$ is unstable, the corresponding $\mathcal{H}_2$ norm becomes infinite and the optimization problem (24a)-(24c) can not be solved. Considering the definition of the $\mathcal{H}_2$ norm for asymptotically stable systems, it becomes maximum iff the integral over the (squared) magnitude of the frequency response becomes a maximum. For an unstable mode, this integral can also be computed by exploiting the fact that the magnitude is not affected when mirroring the unstable pole(s) across the imaginary axis. As a result, an asymptotically stable system is obtained for which the $\mathcal{H}_2$ norm can easily be computed. Based on that, it is proposed to design the blending vectors of an unstable mode by first mirroring the underlying poles across the imaginary axis and then applying the algorithm described in [16]. Note that in order to preserve the magnitude of the frequency response when mirroring a pole, the zeros of each individual transfer channel need to be preserved which typically affects the corresponding residue(s).

For the given aeroelastic model, the $\mathcal{H}_2$-optimal blending vectors may be computed at different trim points and linearely interpolated inbetween. In case the shape of the considered aeroelastic mode changes only marginally for the considered trim points, the corresponding blending vectors also differ only marginally. Since this is the case for the flutter demonstrator described above, the blending vectors are computed at one single trim point and held constant within the whole flight envelope.

### C. Scheduled SISO Controller Design

The free parameters of the SISO controllers $c_j$, i.e. the controller gains, are chosen herein using a model-based multi-objective optimization approach. Therefore, a robust control design problem is defined to tackle the dependency of the underlying aircraft model on the airspeed. The presented model-based gain optimization poses a non-convex design problem, originating from the structured controller setup, which is solved using Matlab's `systune` routine based on non-smooth optimization techniques [17]. The software allows to define criteria either in the frequency domain (e.g., bandwidth) or in the time domain (e.g. rise time) either as minimization criteria (soft constraint) or as minimization constraint (hard criteria).

The multi-model, multi-objective optimization problem to derive a scheduled controller of a predefined structure is derived in [10]. The free parameters in $c_j$ are dependent on the scheduling variables described in the vector $\pi$. This vector belongs to the bounded region $\Pi \in \mathcal{P}$, where $\mathcal{P}$ is the $n_p$-dimensional parameter space. The actual control problem is then defined similarly as in (12), together with the proposed basis selection depending on the indicated airspeed. Herein, classical phase and gain margins are used as criteria in order to improve robustness of the flutter suppression controller. Based on that, the free controller parameters are optimized for each SISO controller $c_j$ dedicated to control the aeroelastic mode $j$.

### D. Simulation Results

For flutter suppression controller design, the aeroelastic model from [15] is trimmed at steady horizontal flight for different airspeeds and linearized around the corresponding trim points. The resulting gridded LTI models consist of $n_x = 1152$ states and are decomposed to obtain the two flutter modes to be controlled. For stabilization of the flutter modes, the $n_u = 2$ outer aileron deflection inputs and the $n_y = 26$ vertical acceleration + pitch rate measurements are used and blended according to Section IV-B. The blending vectors are computed at $V_{\text{ias}} = 60\,\text{m/s}$ and held constant within the whole flight envelope. Applying the computed blends, the symmetric and anti-symmetric flutter modes are well decoupled from each other as depicted in Figure 16. Furthermore, it can be seen that the flutter modes are not fully decoupled from the remaining aeroelastic modes and therefore, a low-pass filter is applied before starting the SISO controller design. The chosen SISO controller structure is of first order for the anti-symmetric flutter mode and of second order for the symmetric flutter mode. Both SISO controllers are scheduled linearely with $V_{\text{ias}}$ and tuned according to Section IV-C, where the resuling controllers are depicted in Figure 17. Closing the loop greatly stabilizes the two flutter modes as it can be seen in the pole migration plot in Figure 18. While the symmetric flutter mode can be stabilized up to an airspeed of 65.5 m/s, the anti-symmetric flutter mode is stabilized even beyond the considered airspeed range. Considering also robustness margins, the maximum airspeed where classical gain and phase margins of at least 6 dB and 45 deg are fulfilled is 60 m/s for both flutter modes. This still means an increase in flutter speed of more than 15 %, see Table 1 for more details. In order to verify the derived flutter suppression controller, nonlinear simulations are performed where aircraft velocity is increased stepwise. As depicted in Figure 19, the flutter speed is increased from 55 m/s to 68 m/s, which means an increase of 23 %.

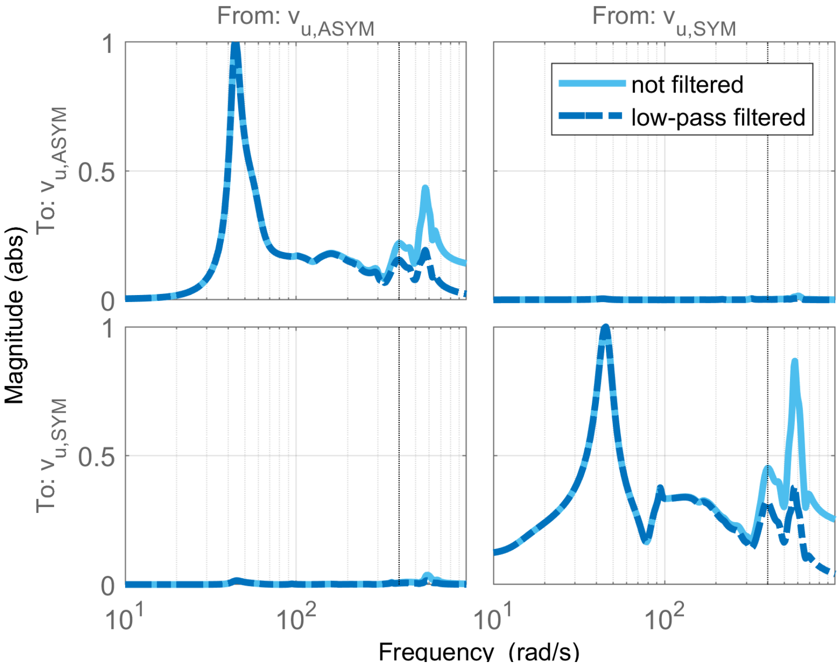


**Fig. 16 Transfer functions from blended inputs to blended outputs indicating a good modal decoupling at $\mathbf{V_{ias} = 20\ m/s}$.**

**Table 1 Comparison of open-loop and closed-loop flutter speed.**

| flutter mode | flutter speed (m/s) | | |
|---|---|---|---|
| | open-loop | closed-loop<br>(no margin) | closed-loop<br>(>45°/6dB) |
| anti-symmetric | 54 | >70 | 60 |
| symmetric | 52 | 66 | 60 |

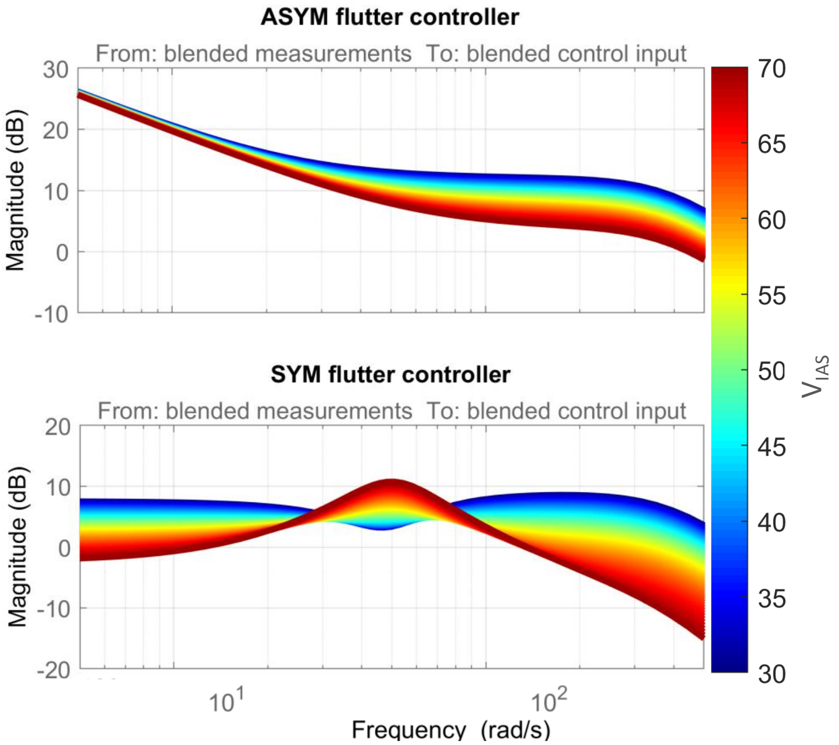


**Fig. 17 Gain-scheduled SISO controllers for aeroelastic mode control.**

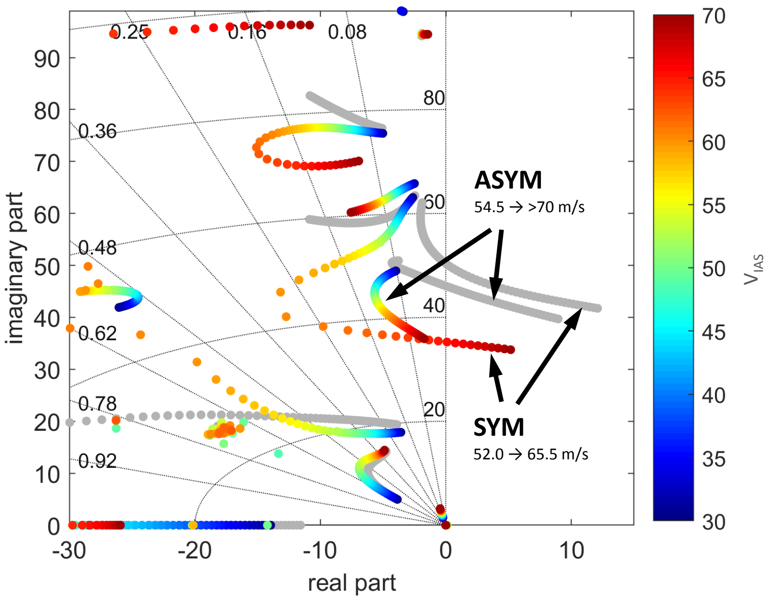


**Fig. 18 Comparison of open-loop (gray) and closed-loop poles (colored).**

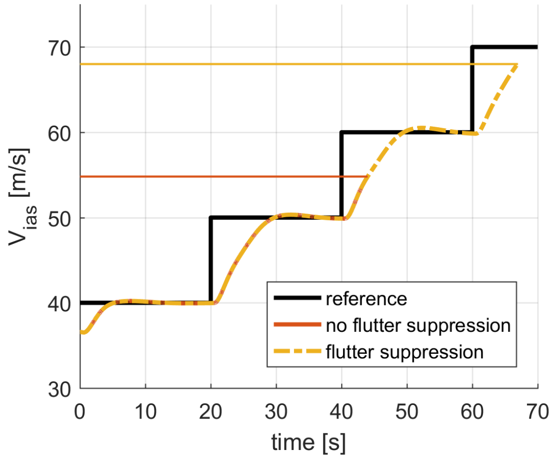


**Fig. 19 Increasing aircraft velocity in nonlinear simulation.**

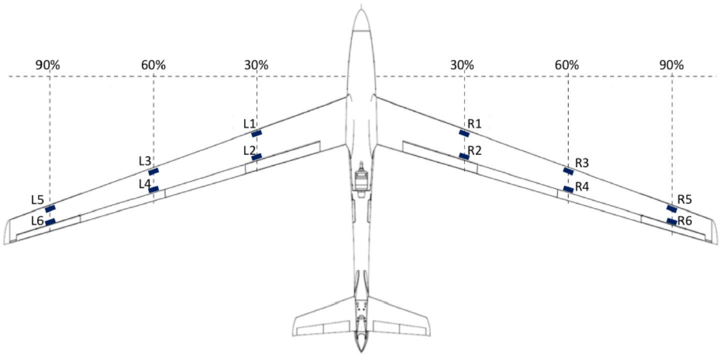


**Fig. 20 FLEXOP sensors for flutter control.**

# V. Robust Flutter Controller Design and Analysis

A second approach utilizing robust control design and analysis techniques has been also developed for the FLEXOP demonstrator as discussed in the Section.

## A. Models for design

The nonlinear model described in Section II was trimmed and linearized in cruise level flight around 26 speeds ($V_{\mathrm{TAS}}$), from 45 up to 70 m/s. From these, a set of 38-state models has been obtained using balanced reduction. The same state transformation matrix is used for all of the models to ensure consistency across speeds.

As discussed the outboard flaps ($\delta_{\text{ail-L4}}$ and $\delta_{\text{ail-R4}}$) are reserved for flutter control, while the midboard flaps ($\delta_{\text{ail-L}\bullet}$ and $\delta_{\text{ail-R}\bullet}$, with $\bullet \in \{2, 3\}$) are used by the baseline controller (see Figure 2. By doing so, the flutter control effort is expected to be decoupled from the guidance action, thus minimizing its effect on the dynamics as seen by the (auto)pilot. Since the effect of both symmetric and antisymmetric flutter modes must be countered, we shall ensure that both actuators are controlled separately.

Each wing of the demonstrator is equipped with 6 IMUs, as indicated in Figure 20. We shall use the outboard sensors ($L5$, $L6$, $R5$ and $R6$), together with the IMU on the center of gravity, to provide measurements for the flutter controller. The measured outputs used by the controller are the pitch rate $q$, the center of gravity acceleration $a_{z-\mathrm{cg}}$ and the left and right wingtip accelerations, $a_{z-\mathrm{wL}}$ and $a_{z-\mathrm{wR}}$ respectively. We use the mean between the measurements provided by sensors 5 and 6 to define the acceleration on each wingtip.

The delay introduced by the computation of the control law as well as the servo controller is estimated to be of 10 ms, which is modeled as a pure delay $G_{\mathrm{delay}}(s) = e^{-0.01s}$.

### B. Controller design

In this section, we describe the control design approach used to synthesize the flutter suppression controller. Two different controllers as considered, first a classic $H_\infty$ controller, which is then used as a starting point for the design of a structured $H_\infty$ controller.

#### *1. $H_\infty$ controller*

The design strategy used to synthesize the controller builds up on the experience reported in [18]. The controller is designed using the reduced-order model at a cruise speed of 57 m/s, which is beyond the flutter speed. In this configuration, the eigen-frequencies of the unstable symmetric and antisymmetric flutter modes are 44.9 rad/s and 42.5 rad/s.

The goal of the controller is to provide damping to the flutter oscillations. In order to translate this specification into an $H_\infty$ optimization problem, we choose as performance measures the derivative of the generalized coordinates corresponding to the flutter modes, $\dot{\eta}_1$ and $\dot{\eta}_2$. This results in a transfer matrix having resonance peaks at both of the flutter modes' eigen-frequencies. The damping objective can then be translated as a desired upper bound on these transfer functions.

In $H_\infty$ control, the designed controller will have as many states as the size of the synthesis model. In order to reduce the plant size, we obtain a reduced-order model of the combined effects of the delay with the IMU and actuator dynamics. For this, we use a fifth-order Padé approximation of the pure delays, and the obtained model is reduced via balancing and residualization. The result is a third-order model that very closely represents the phase loss of the parasitic dynamics. Adding this reduced actuator and sensor model to the aircraft' inputs, we end up with a synthesis model $G(s)$ with two inputs and 6 outputs, see Figure 21.

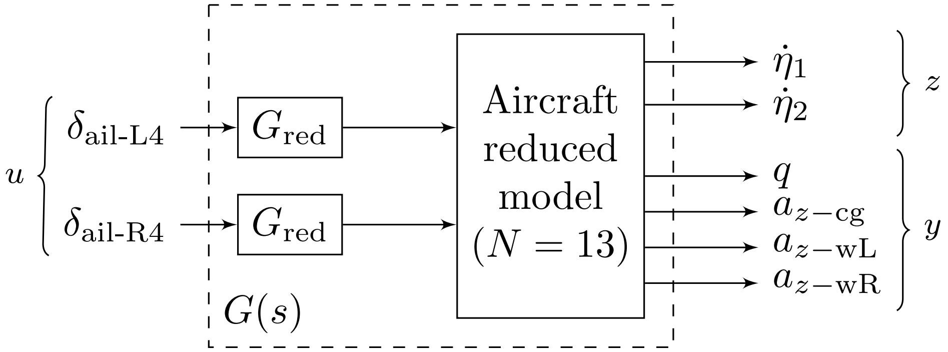


**Fig. 21 Synthesis model for $H_\infty$ control design.**

The plant $G(s)$ is put into the standard $H_\infty$ problem form, see Figure 22. The weights are added to translate the specifications of the control design into the $H_\infty$ optimization problem. The damping provided by the controller for the unstable flutter modes is specified via the weight $W_z$. By increasing its value, we are imposing a lower upper bound on the resonance peak of the respective transfer functions, thus increasing the flutter damping. The weight $W_u$ is chosen so as to confine the control action around the flutter eigenfrequencies. This is done to avoid exciting high frequency aeroelastic modes as well as limiting the coupling with the low-frequency dynamics that are handled by the baseline controller. We then choose the same weighting function for each control channel, given by

$$\tilde{W}_u(s) = 100\frac{s^2 + 101s + 2200}{s^2 + 127020s + 2200}. \tag{25}$$

This transfer function configures a band-stop filter centered around the eigenfrequency of the flutter modes, which imposes a wash-out and roll-off effect on the controller's frequency response, thus achieving the desired behavior. The other weighting functions are chosen as constant matrices whose goal is to scale the input and output channels. The values of the weights chosen after tuning are:

$$\begin{aligned} W_d &= I_2 & W_z &= 0.01\,\text{diag}(2, 8) \\ W_n &= 200\,\text{diag}(.055, 0.75, 1, 1) & W_y &= 0.0001\,\text{diag}(2000, 50, 2.5, 2.5) \\ & & W_u(s) &= \text{diag}\left(\tilde{W}_u(s), \tilde{W}_u(s)\right) \end{aligned} \tag{26}$$

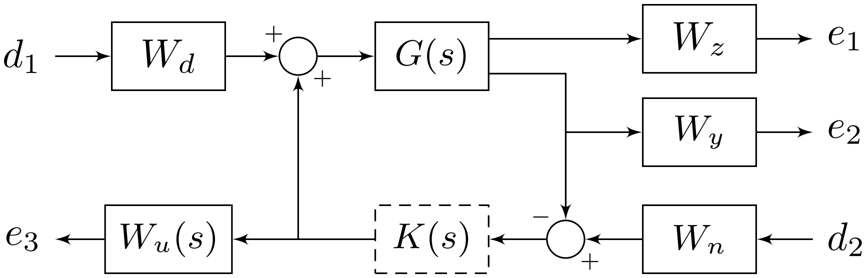


**Fig. 22 Generalized plant for $H_\infty$ optimization.**

The synthesis model has 48 states, 38 coming from the reduced-order models, 6 from the reduced parasitic dynamics $G_{\text{red}}$ and additional 4 from $W_u$. The system augmented with the weights is put in a standard $H_\infty$ formulation, and the controller is then found by solving the associated $H_\infty$ optimization problem. In view of the need for fast computation of the control law with limited processing power by the flight computer, we perform a model reduction by truncating a balanced representation of the controller, to arrive at a reduced version with 4 states. Although the frequency responses differ at low and high frequencies, they are quite similar around the flutter eigenfrequencies, where the controller needs to function.

A pole-zero map of the reduced-order closed-loop system with the reduced controller is given in Figure 23 (with a fifth-order Padé approximation of the open-loop input delay). The controller stabilizes the closed-loop system for flying configurations from $N = 1$ up to $N = 21$.

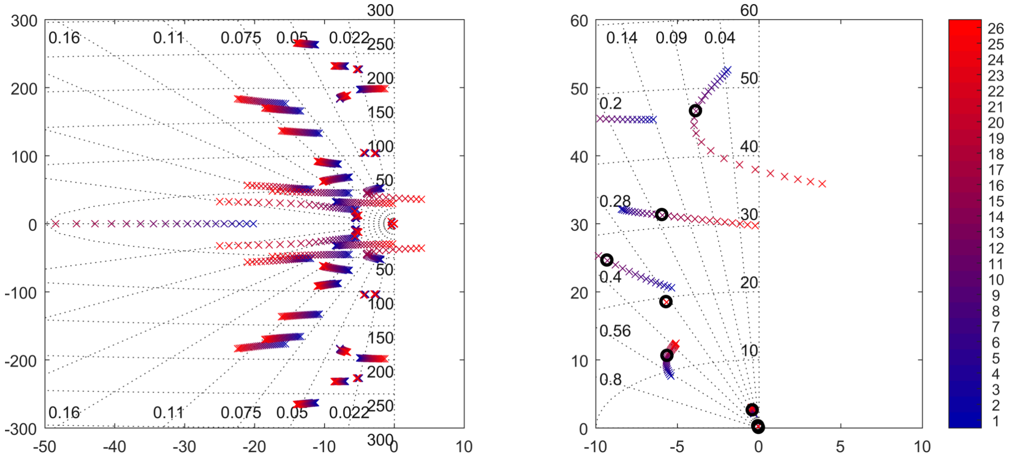


**Fig. 23 Pole-zero map of the fifth order Padé approximation of the reduced-order closed-loop system using the reduced controller, with emphasis on the poles at the synthesis configuration ($N = 13$, $V_{\text{TAS}} = 57$ m/s)**

The analysis of the performance improvement achieved by the flutter control strategy is done with the 1152-state full flexible linearized models. We begin by analyzing the closed-loop in cruise flight at the same speed as the synthesis model ($N = 13$, $u \approx 57$ m/s). The maximum singular value from the inputs $\delta_{4L}$ and $\delta_{4R}$ to the outputs $\dot{\eta}_1$ and $\dot{\eta}_2$ for both the open-loop and closed-loop systems are shown in Figure 24. It shows how the flutter controller damps the flutter modes without increasing the gain for the low and high-frequency dynamics.

Figure 25 shows the results of a step input in the elevator. We can see that the flutter controller has little impact on the maneuverability of the aircraft, and the rigid body dynamics remain virtually the same as the open-loop ones. This indicates that the flutter controller should not interfere with the baseline control. Above the flutter speed, we see that the controller stabilizes the flutter modes, and Figure 25 shows that the flutter oscillations are effectively damped within 1 second.

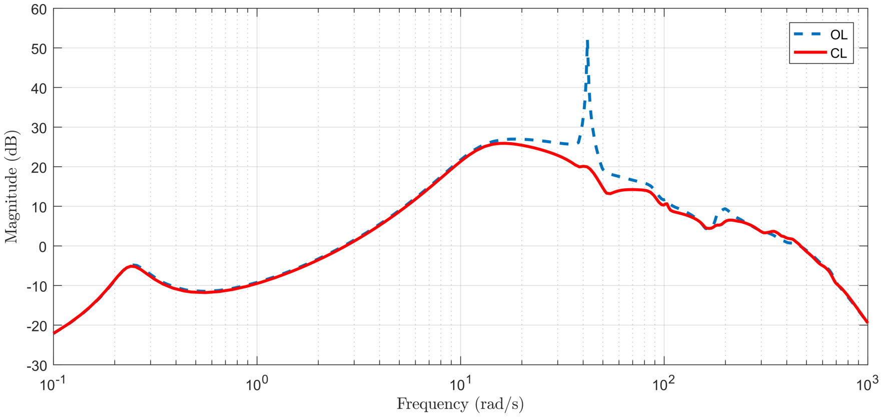


**Fig. 24 Maximum singular value of open-loop and closed-loop systems at the speed chosen for synthesis ($N = 13$) between the inputs $\delta_{4L}$ and $\delta_{4R}$ and the performance outputs $\dot{\eta}_1$ and $\dot{\eta}_2$.**

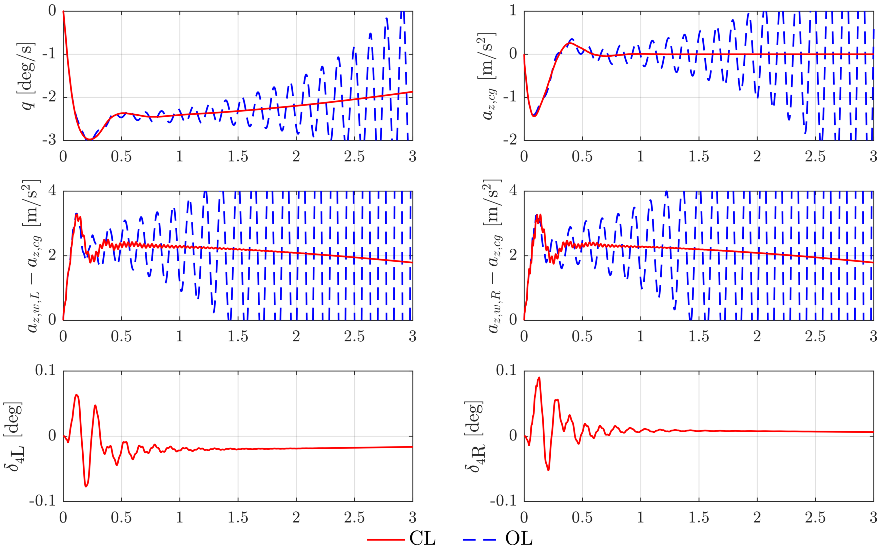


**Fig. 25 Response of the closed-loop system with $K_r$ to a step in the elevator $\delta_e$ of the open-loop and closed-loop for $N = 10$ (above the open-loop flutter speed).**

### *2. Structured $H_\infty$ controller*

We now proceed to perform a different controller synthesis using structured $H_\infty$ control [19, 20]. This technique uses non-smooth optimization tools to synthesize a controller minimizing an $H_\infty$ metric, and allows for the consideration of additional structural constraints on the controller such as fixed order or fixed structure. Another advantage of this approach is the possibility to synthesize multi-plant and/or auto-scheduled controllers. Although the underlying problem is non-convex and NP-hard, the optimization procedure is shown to always converge to a local minimum [19]. Moreover, it has been shown to yield good results in a wide range of applications, including space missions and aircraft control [21].

The idea is to try to improve on the $H_\infty$ synthesis performed in Section 1, where a 4th-order controller $K_r$ was obtained through a balanced reduction. For the new synthesis, we enforce a a 4th order structure on the controller, and we use the reduced controller $K_r$ as an initial condition for the optimization. The synthesis is performed using the same standard $H_\infty$ formulation and with the same weights as in the previous section, see (26).

The controller obtained via this approach is denoted $K_s$. The comparison between the frequency response of the original reduced-order $H_\infty$ controller $K_r$ and the structured $H_\infty$ controller $K_s$ showed that although the frequency responses related to the acceleration inputs are relatively similar, the structured controller has a higher gain in the pitch feedback channel. In addition, the wash-out effect is improved in the new controller, while the roll-off is somewhat degraded.

Let us now consider the analysis of the closed-loop system using the controller $K_s$. We begin by analyzing the pole placement of the closed-loop system made of the Padé approximation of the reduced-order synthesis models and the structured controller $K_s$. We see that the controller stabilizes the symmetric mode for all flight configurations, while the asymmetric mode remains unstable for $N \geq 19$. Compared to $K_r$, this means a reduced envelope expansion.

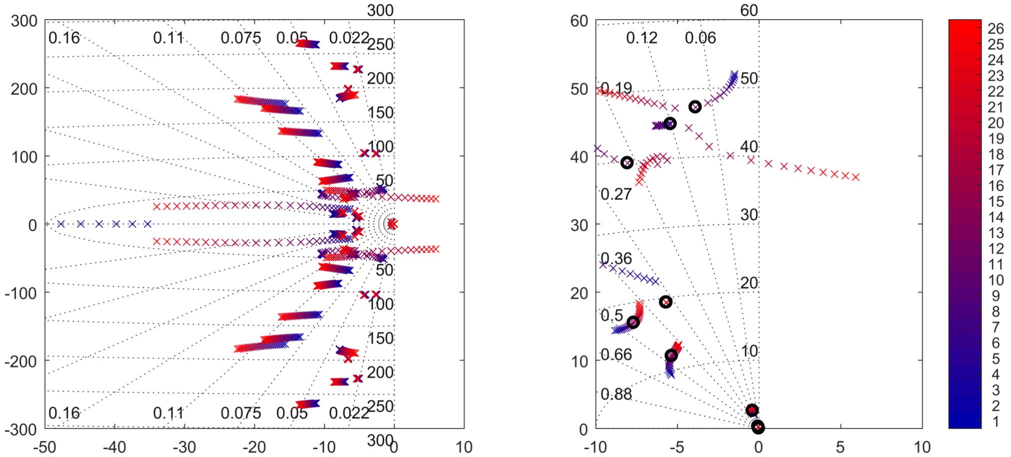


**Fig. 26 Pole-zero map of the fifth order Padé approximation of the reduced-order closed-loop system using the structured $H_\infty$ controller, with emphasis on the poles at the synthesis configuration ($N$ = 13, $V_{\text{TAS}}$ = 57 m/s)**

As we did in the previous section, we analyze the closed-loop controller using the full-order 1152-state linearized models. Figure 27 shows the maximum singular value between the inputs $\delta_{\text{ail-L4}}$ and $\delta_{\text{ail-R4}}$ and the outputs $\dot{\eta}_1$ and $\dot{\eta}_2$ in the synthesis flight configuration ($N = 13$). It appears that the new controller $K_s$ provides a little less damping for the flutter modes.

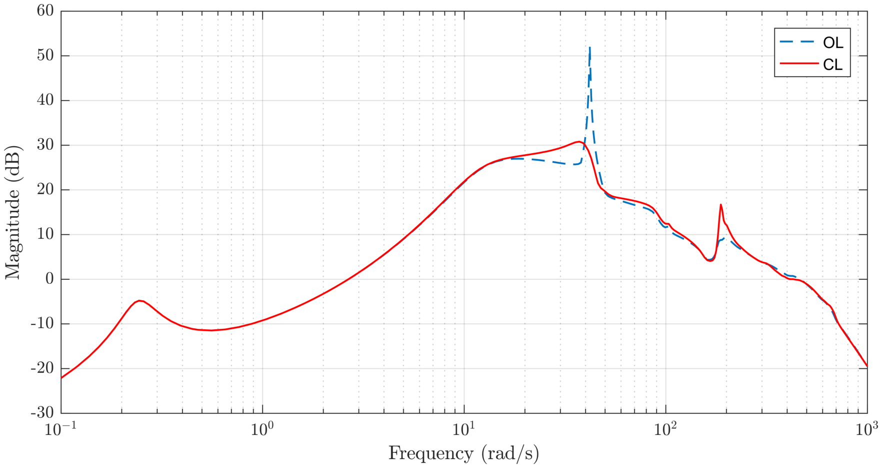


**Fig. 27 Maximum singular value of open-loop and closed-loop systems at the speed chosen for synthesis ($N$ = 13) between the inputs $\delta_{4L}$ and $\delta_{4R}$ and the performance outputs $\dot{\eta}_1$ and $\dot{\eta}_2$.**

Figure 28 shows the response of the closed-loop system to a step in the elevator input. The structured controller performs well, and damps the flutter oscillations in less than 1 second. The presence of high-frequency oscillations is less pronounced than the results obtained with $K_r$, see Figure 25.

### C. Worst-case LTV analysis

It is clear from the previous discussions that for aeroelastic systems the properties of the response are markedly dominated by the flight speed $V$. For these reasons, aircraft maneuvers involving a change in speed are inherently

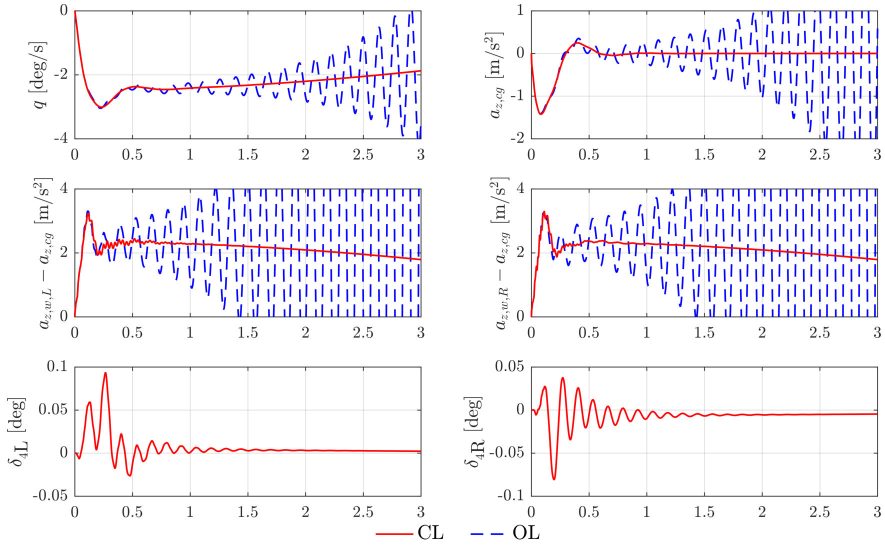


**Fig. 28 Response of the closed-loop system with $K_s$ to a step in the elevator $\delta_e$ of the open-loop and closed-loop for $N = 10$ (below the flutter speed).**

time-varying. Due to the emphasis of the FLEXOP project on flight tests, it is thus deemed important to take into account this by considering a mathematical description of the aircraft dynamics as Linear Time-Varying (LTV) .

Prompted by these observations, the application of the worst-case LTV analysis framework from [22] to the controllers described in Sec. V-B is reported here. Specifically, the susceptibility of the demonstrator to atmospheric gust during a notional finite-horizon flight test maneuver is investigated.

### *1. Results*

The LTV framework was applied to the open loop and closed loop FLEXOP demonstrator model in order to gain further insights on the flutter behavior and effectiveness of the active flutter control solutions proposed by the University of Bristol team.

The scenario considered consists of a uniformly accelerated level flight manoeuvre from a speed $V_1$ to $V_2$. The manoeuvre starts at $t = 0$ with an initial speed $V_1$ and is concluded at $t = T = 2\frac{L_{ac}}{V_1+V_2}$, where $L_{ac}$ denotes the covered distance. Unless otherwise specified, it will be considered $V_1 = 45\frac{m}{s}$, $V_2 = 49\frac{m}{s}$, and $L_{ac} = 250m$ (hence $T = 5.3s$).

The plant is described by a grid of LTI plants obtained at 5 uniformly spaced points between $V_1$ and $V_2$. The models have 38 states and are the same described for control design in Sec. V-A. By assuming an uniformly accelerated manoeuvre, the state-matrices can be interpolated with respect to time in the horizon $[0, T]$. This allows to finally build up the LTV model $G$, capturing the variability of the aircraft properties in the speed interval $[V_1, V_2]$.

The disturbance $d$ assumed here is a uniform vertical wind gust, accounted for by means of the 8 control surface inputs (i.e. $\delta_{\text{ail-L}\bullet}$ and $\delta_{\text{ail-R}\bullet}$, with $\bullet = 1, 2, 3, 4$). The premise for this is that, to a first approximation, the effect of a vertical gust is to change the local angle of attack of the wing, thus it can be captured as an equivalent rotation of the control surfaces. A uniform symmetric gust is considered here, thus the control surfaces have all the same rotation, i.e. $\delta_{\text{ail-L}\bullet} = \delta_{\text{ail-R}\bullet} = d$. As for the outputs $e$, two different cases will be studied: vertical acceleration at the tip of the right wing $a_{z-\text{tR}}$ (specifically, at the sensor R6 location) and at the aircraft center of gravity (CG) $a_{z-\text{CG}}$. In both cases, $e$ is normalized with the gravitational acceleration $g$. Note finally that, due to the linearity of the problem, the worst-case disturbance can be arbitrarily scaled. For a better representation and comparison of the time-domain responses, the signals shown in the plots are adimensionalized and normalized such that $||d||_{2,[0,T]} = 1$.

The objective of the analyses here is to investigate the difference between open and closed-loop performance during a finite-horizon manoeuvre. Other aspects, including the importance of capturing the time-varying nature of the problem and a comparison of this approach with traditional gust analyses employed in the aerospace community, were explored in [22].

For the closed-loop case, both the controllers designed in Sec. V-B are considered. Table 2 shows a comparison based on the $\mathcal{L}_2$-gain for: open loop (OL); closed loop with standard $\mathcal{H}_\infty$ design (CL-1); closed loop with structured

$\mathcal{H}_\infty$ (CL-2).

**Table 2 Comparison of OL and CL performance**

| Plant | $\|\|G\|\|_2^{tR}$ | $\|\|G\|\|_2^{CG}$ |
|---|---|---|
| $OL$ | 20 | 2.7 |
| $CL-1$ | 4.3 | 0.72 |
| $CL-2$ | 4.8 | 0.75 |

It is first seen that both the controllers are able to significantly reduce the gains. It can also be noticed that the gap between $||G||_2^{tR}$ and $||G||_2^{CG}$, which can be interpreted as a measure of the flexibility of the wing, is reduced. Thus, the analyses showcase the ability of the controller to tackle it. Moreover, the analyses point out that the standard $\mathcal{H}_\infty$ controller performs slightly better in the analyzed cases.

Complementing the quantitative information from Table 2, it is possible to compute the worst-case disturbance that maximizes the $\mathcal{L}_2$-gain of the selected output. Fig. 29 shows a comparison between the open-loop and closed-loop cases for the CG acceleration.

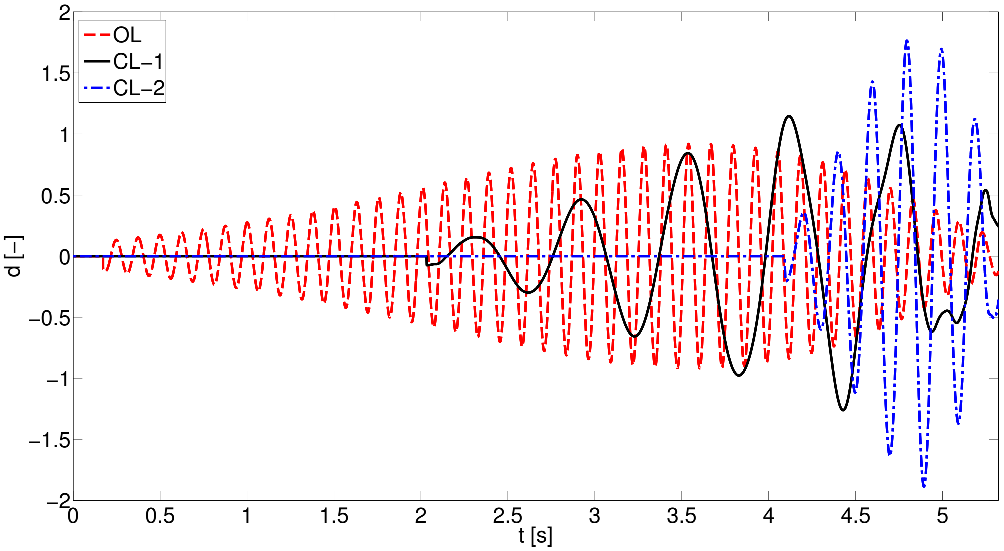
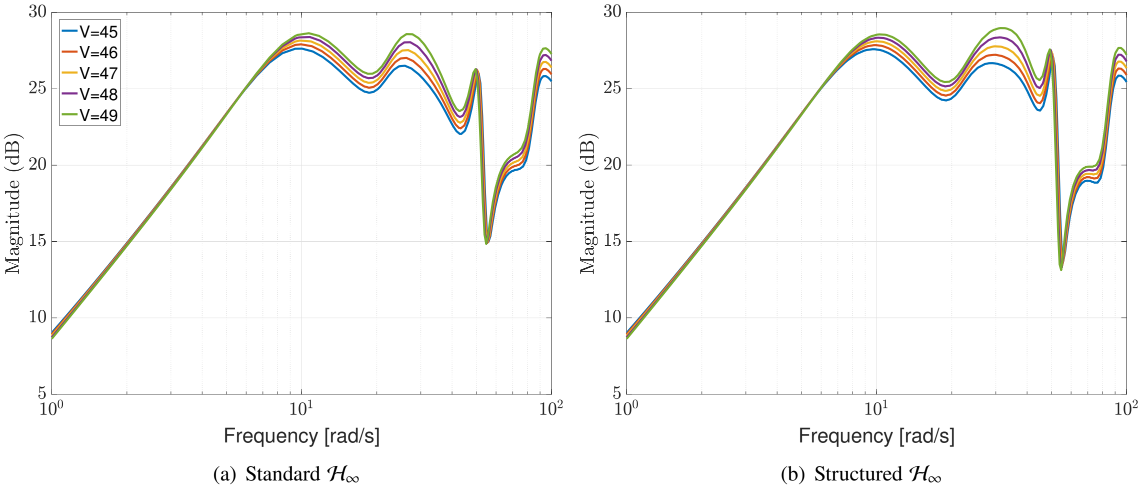


**Fig. 29 $\mathcal{L}_2$ gain worst-case disturbances for $a_{z-\mathrm{CG}}$.**

A substantial difference is observed in the (dominant) frequency of the signals. The $OL$ has a frequency of about $50\frac{rad}{s}$, close to that of the two first bending modes (recall previous discussions), whereas the closed loop responses have lower ones. This shows that the controller achieves a reduction in the closed-loop $\mathcal{L}_2$ gain by reducing the energy associated with the first elastic modes. Specifically, *CL-1* and *CL-2* have frequencies of approximatively $10\frac{rad}{s}$ and $30\frac{rad}{s}$, respectively. Another feature apparent from Fig. 29 is that the disturbance acts on a shorter time window for the closed loop cases (with the shortest registered for *CL-2*).

To interpret these results, Fig. 30 shows the Bode plots (magnitude only) for the two closed loop systems at 5 uniformly spaced speeds between $V_1$ and $V_2$.

The plot allows to clearly appreciate two peaks at $10\frac{rad}{s}$ and $30\frac{rad}{s}$, having a larger value as speeds is increased. Two associated features are worth highlighting. First, it can be observed that the larger peak for the Standard $\mathcal{H}_\infty$ design is the low frequency one ($10\frac{rad}{s}$), while for the Structured $\mathcal{H}_\infty$ design is the higher one ($30\frac{rad}{s}$). This could in turn justify the different frequencies observed in Fig. 29. Second, it can be noted that for the Structured $\mathcal{H}_\infty$ design (Fig. 30(b)) there is a larger variability in the curves depending on the value of the speed (particularly for the peak at $30\frac{rad}{s}$) when compared to the Standard $\mathcal{H}_\infty$ design. This could motivate why the closed-loop worst-case disturbance signals acted on a shorter time window in Fig. 29. Indeed the aircraft is uniformly accelerating, thus it is more *advantageous*



(a) Standard $\mathcal{H}_\infty$

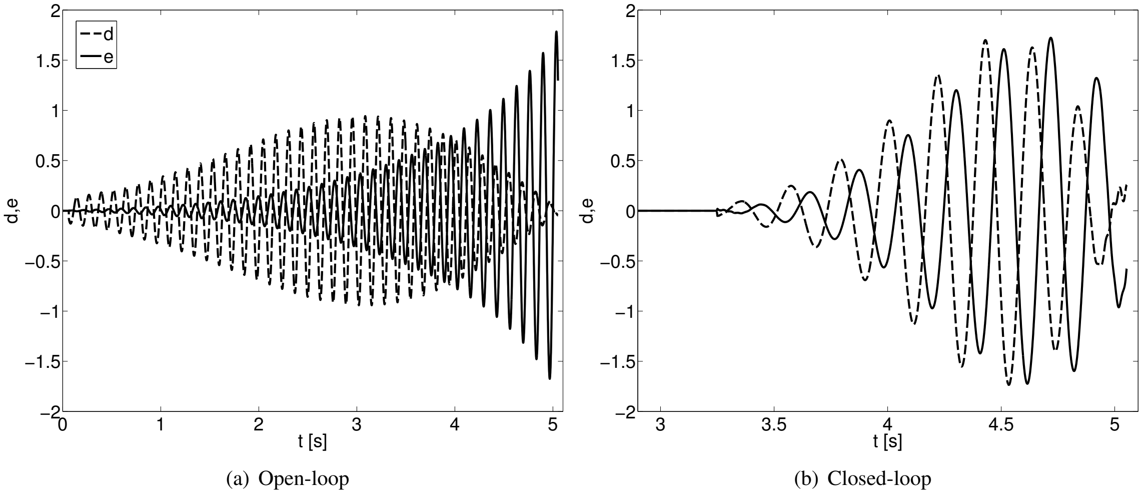
(b) Structured $\mathcal{H}_\infty$

**Fig. 30 Bode (magnitude) plot from the gust channel to $a_{z-\mathrm{CG}}$.**

for the disturbance to act towards the end of the maneuver (where the speed is larger) in order to maximize the extracted energy. In view of the previous observation on the effect of the speed on the Bode plot, this rationale applies even more for the Structured $\mathcal{H}_\infty$ case, and this can motivate the trend observed in Fig. 29 for *CL-2*.

In order to assess the effect of using a controller to suppress flutter, the final airspeed of the manoeuvre $V_2$ is increased from $49\frac{m}{s}$ to $54\frac{m}{s}$. This airspeed is beyond the open-loop flutter speed, thus the aircraft could not be flown here without flutter suppression strategies. The case of the open-loop plant closed with the Standard $\mathcal{H}_\infty$ controller is considered here. The frozen LTI analyses in Sec. V-B showed that the system is stabilized up to an airspeed $V = 65\frac{m}{s}$. It is therefore of interest to analyze the modified trajectory within the LTV framework.

Fig. 31 reports the worst-case disturbance and the output $e = a_{z-\mathrm{tR}}$ , obtained via simulation of the OL (Fig. 31(a)) and CL (Fig. 31(b)) LTV systems. For the sake of clarity in comparing the signals, the acceleration is dimensionalized such that $||a_{z-\mathrm{tR}}||_{2,[0,T]} = 1$.



(a) Open-loop

(b) Closed-loop

**Fig. 31 Worst-case disturbance ($d$) and corresponding wing tip acceleration ($e$) for a trajectory with $V_2$ larger than the open-loop flutter speed.**

The open-loop response, featuring a finite horizon gain $||G||_2^{tR} = 58.8$ (note the drastic increase compared to 20 in

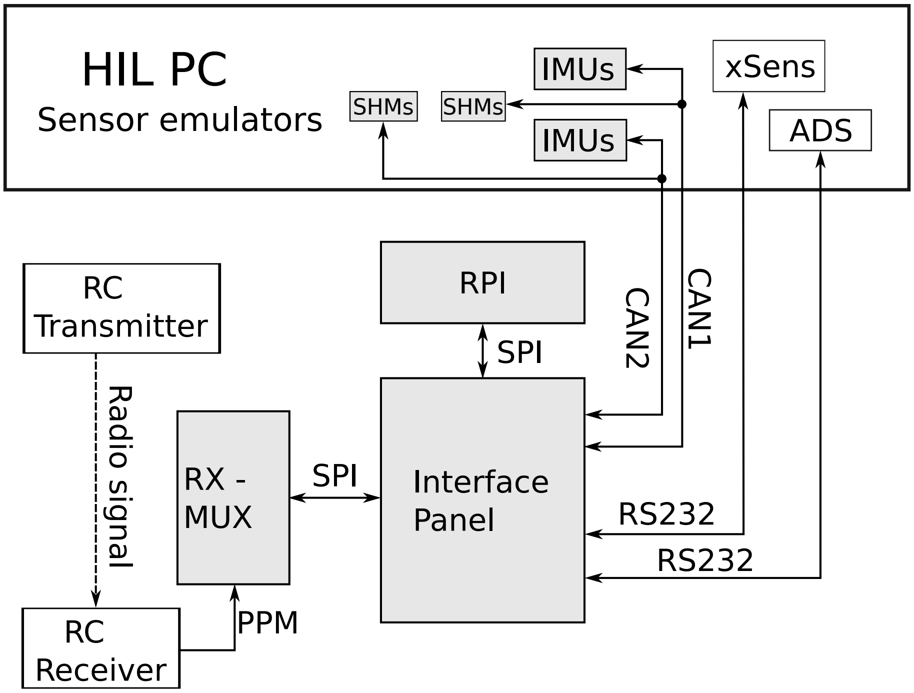


**Fig. 32 An over schema of the connection and setup of the HIL developed for the FELXOP demonstrator**

Tab. 2), exhibits an unstable behavior. Indeed, by looking at the second part of the response, it can be seen that energy is extracted from the system even when the input signal is decaying. Note instead that in the closed-loop the excitation coming from the gust is damped out. Moreover, a performance of $||G||_2^{tR} = 5.34$, which is only slightly larger than the gain 4.3 featured with the other trajectory ($V_2 = 49\frac{m}{s}$), is registered.

## VI. Hardware In the Loop testing

Before the first flight testing of the discussed controllers, an in-depth hardware in the loop (HIL) testing has to be carried out. Due to the many physical actuators and different sensors used in the FLEXOP demonstrator aircraft, off the self flight computer (FCC) solutions do not have enough functions and connections, therefore developing a reliable testing environment is necessary. In addition, due to the unique system specifications and its environment, off the self solutions for our HIL is not usable either. Therefore, a HIL environment was developed for testing the controllers, before real flights. The overall interconnection of the HIL is depicted in Figure 32.

The FCC itself is based on a Raspberry pi 3B+ embedded computer and two additional board developed for this project. The Raspberry pi runs a modified OS, which has real time capabilities, and an user application, which is responsible for data logging and autopilot functions. In connection with the FCC, a FlighHAT serves as an all purpose interface card, and two RX-MUX units, which main role is to give reference signal to the actuators. All the sensors are connected to the FlightHAT board: it collects every data from the system, and send them to the Raspberry pi. The xSens and the uADS sensors has RS-232 interface, and the IMU and SHM units designed specifically for the FLEXIOP demonstrator, with added CAN interfaces. The main role of the RX-MUX units is to convert RC PPM signals into actuator reference signals, and to switch between autopilot and manual pilot modes. The FlighHAT also communicates with the RX-MUX units, to be able to send new reference signals generated by the autopilot software and store data generated by the RX-MUX unit.

The main concept of the HIL simulation is to test the flight control computer. With such a simulation environment, the FCC hardware and the autopilot and flutter controller softwares are also testable. For proper simulations, the system should work as if it would fly, so the values, which were originally measured by the sensors, have to be emulated by the simulation itself. This means, all the IMUs, SHMs, the xSens and the uADS need to be replaced by sensor emulators.

The emulators need to behave exactly the same, as the original sensors (same message structure, same communication protocol, same timing and frequency, etc.).

The sensor emulators are connected to the FCC via the same connectors, which are actually used for measurements during the flight. For proper working of the HIL, the actuator values need to be sent to the simulator PC. This is done by the FCC using the same buses, as the sensor emulators use. With that the FCC works the same way in simulated and real environment as well.

Finally, the entire FLEXOP nonlinear model and the sensor emulators are placed into a MATLAB/Simulink file. For visualizing the HIL test, it is able to put scopes into the Simulink model, but the simulation is also connected to the FlightGear software. This software visualizes the orientation and the position of the airplane.

## VII. Conclusions and Further Steps

In the paper we have presented the control design aspects of a flexible demonstrator UAV for the FLEXOP project. First, the control oriented modeling is discussed, followed by the rigid body control designs. Two flutter suppression controller methodologies are also developed by using the dedicated control surfaces. Software simulations are reported, with proved capability to extend the flight envelope of the aircraft. Hardware in the loop simulations are currently evaluated before the first flight tests in February 2019.

**Acknowledgments**

This work was performed in the framework of the European Union's Horizon 2020 research and innovation programme and is part of the Flutter Free FLight Envelope eXpansion for ecOnomic Performance improvement (FLEXOP) project with the grant agreement No 636307.

## References

[1] Stahl, P., Sendner, F.-M., Hermanutz, A., Rößler, C., and Hornung, M., "Mission and Aircraft Design of FLEXOP Unmanned Flying Demonstrator to Test Flutter Suppression within Visual Line of Sight," *Proc. of 17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017.

[2] Sendner, F.-M., Stahl, P., Rößler, C., and Hornung, M., "Designing an UAV Propulsion System for Dedicated Acceleration and Deceleration Requirements," *Proc. of 17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017.

[3] Roessler, C., Stahl, P., Sendner, F., Hermanutz, A., S. J. Koeberle, J. B., Rosov, V., Breitsamter, C., Hornung, M., Meddaikar, Y. M., Dillinger, J. K. S., Sodja, J., de Breuker, R., C. Koimtzoglou, D. K., and Georgopoulos, P., "Aircraft Design and Testing of FLEXOP Unmanned Flying Demonstrator to Test Load Alleviation and Flutter Suppression of High Aspect Ratio Flexible Wings," *AIAA Scitech Forum and Exposition*, 2019, pp. –.

[4] Meddaikar, Y. M., Dillinger, J. K. S., Klimmek, T., Krüger, W., Wüstenhagen, M., Kier, T., Hermanutz, A., Hornung, M., Rozov, V., Breitsamter, C., Alderman, J., Takarics, B., and Vanek, B., "Aircraft Modelling Toolchain for the FLEXOP Unmanned Flying Demonstrator," *AIAA Scitech Forum and Exposition*, 2019, pp. –.

[5] Wuestenhagen, M., Kier, T., Meddaikar, Y. M., Pusch, M., Ossmann, D., and Hermanutz, A., "Aeroservoelastic Modeling and Analysis of a Highly Flexible Flutter Demonstrator," *2018 Atmospheric Flight Mechanics Conference*, 2018, p. 3150.

[6] Hofstee, J., Kier, T., Cerulli, C., and Looye, G., "A variable, fully flexible dynamic response tool for special investigations (VarLoads)," *International Forum on Aeroelasticity and Structural Dynamics*, 2003.

[7] Kier, T., and Looye, G., "Unifying manoeuvre and gust loads analysis models," *International Forum on Aeroelasticity and Structural Dynamics*, American Institute of Aeronautics and Astronautics, 2009.

[8] Tewari, A., *Aeroservoelasticity: Modeling and Control*, Springer, 2015.

[9] Karpel, M., "Reduced-order aeroelastic models via dynamic residualization," *Journal of Aircraft*, Vol. 27, No. 5, 1990, pp. 449–455.

[10] Apkarian, P., Dao, M.-N., and Noll, D., "Parametric Robust Structured Control Design," *IEEE Transactions on Automatic Control*, 2015.

[11] Apkarian, P., Gahinet, P., and Buhr, C., "Multi-model, multi-objective tuning of fixed-structure controllers," *Proc. of European Control Conference*, Strasbourg, France, 2014.

[12] McRuer, D. T., Graham, D., and Ashkenas, I., *Aircraft Dynamics and Automatic Control*, Vol. 740, Princeton University Press, Princeton, NJ, 2014.

[13] Brockhaus, R., Alles, W., and Luckner, R., *Flugregelung*, 3$^{rd}$ ed., Springer Verlag Berlin Heidelberg, 2011.

[14] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft control and simulation: dynamics, controls design, and autonomous systems*, 2015.

[15] Wüstenhagen, M., Kier, T., Pusch, M., Ossmann, D., Meddaikar, M. Y., and Hermanutz, A., "Aeroservoelastic Modeling and Analysis of a Highly Flexible Flutter Demonstrator," *Proc. of 2018 Atmospheric Flight Mechanics Conference, AIAA AVIATION Forum*, Atlanta, GA, 2018, pp. 1–15.

[16] Pusch, M., "Aeroelastic Mode Control using $\mathcal{H}_2$-optimal Blends for Inputs and Outputs," *AIAA Guidance, Navigation, and Control Conference*, Orlando, FL, USA, 2018, pp. 1–13.

[17] Apkarian, P., and Noll, D., "Nonsmooth $H_\infty$ Synthesis," *IEEE Transactions on Automatic Control*, Vol. 51, No. 1, 2006, pp. 71–86.

[18] Theis, J., Pfifer, H., and Seiler, P. J., "Robust Control Design for Active Flutter Suppression," *AIAA Atmospheric Flight Mechanics Conference*, American Institute of Aeronautics and Astronautics, 2016. doi:10.2514/6.2016-1751.

[19] Apkarian, P., and Noll, D., "Nonsmooth $H_\infty$ synthesis," *IEEE Transactions on Automatic Control*, Vol. 51, No. 1, 2006, pp. 71–86. doi:10.1109/TAC.2005.860290.

[20] Gahinet, P., and Apkarian, P., "Structured $H_\infty$ Synthesis in MATLAB," *IFAC Proceedings Volumes*, Vol. 44, No. 1, 2011, pp. 1435–1440. doi:10.3182/20110828-6-IT-1002.00708, 18th IFAC World Congress.

[21] Marcos, A., and Sato, M., "Flight testing of an structured H-infinity controller: An EU-Japan collaborative experience," *IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 2132–2137. doi:10.1109/CCTA.2017.8062768.

[22] Iannelli, A., Seiler, P., and Marcos, A., "Construction of worst-case disturbances for LTV systems with application to flexible aircraft," *Journal of Guidance, Control, and Dynamics*, 2019.