Comparison of Flight FTC Techniques: from standard and structured H∞, to self-scheduling and LPV

Andrès Marcos, Sergio Waitman, Masayuki Sato

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-35 (2022) 61–66

Comparison of Flight FTC Techniques: from standard and structured H_∞ , to self-scheduling and LPV ⋆

A. Marcos * **S. Waitman** ** **M. Sato** ***

* *Universidad Carlos III de Madrid (UC3M), Madrid, Spain (e-mail: anmarcos@ing.uc3m.es)*

** *ONERA - The French Aerospace Lab, Toulouse, France (e-mail: sergio.waitman@onera.fr)*

*** *Japan Aerospace Exploration Agency (JAXA), Tokyo, Japan, (e-mail: sato.masayuki@jaxa.jp)*

Abstract: In this article several robust control design techniques are compared via their application to the fault tolerant control problem for the lateral/directional motion of JAXA's research aircraft MuPAL-α. The techniques used include: (i) a single, passive-FTC, robust structured H_∞ design, (ii) single, active-FTC, robust standard and structured H_∞ designs, (iii) manual scheduling schemes from the previous designs, (iv) a self-scheduled structured H_∞ design, and (v) a linear parameter varying design. All the designs were implemented in the onboard computer and validated in the so-called Aircraft-In-the-Loop configuration, which entails the operation of the full aircraft in fly-by-wire mode in the hangar. The results show that all the approaches provided acceptable solutions, but with the last two techniques resulting in a more homogeneous performance throughout the fault and command scenarios tested.

Keywords: Robust Control, Fault Tolerant Control, Flight Control

1. INTRODUCTION

The standard practice in flight control is to manually interpolate a series of local, linear controllers designed at specific flight conditions (e.g. airspeed and altitude) into a so-called global controller following the gain-scheduling technique (GS) (Åström and Wittenmar, 1995). In the case of faults of significant criticality, this global controller is switched to another one designed for more robustness, while simultaneously deactivating an increasing number of autopilot functionalities (Goupil, 2011). Typically, the synthesis of the abnormal-situation controller does not take into account fault information, resulting in unnecessarily less performance. This is the result of the GS technique's drawbacks: (1) as the number of parameters to consider increases, there is a much larger number of controllers to design and their similarity (in structure and/or piece-wise value) more difficult to achieve, and (2) both of these issues make the scheduling step unfeasible since it is ad-hoc and manual (thus, only possible for small number of parameters and limited interpolation rules' complexity). Commercial aviation has reduced the risk and effort of GS flight control design by using fixed structures (the longitudinal C* and lateral-directional Y* laws) and by following the aforementioned rationale in case of faults.

There have been intensive efforts from the academic control research community to address the above issues by developing: (1) advanced techniques to design individual controllers with better performance-vs-robustness trade-off (Balas et al., 2015a) and/or a fixed structure (Gahinet and Apkarian, 2011); (2) synthesis techniques directly including fault information (Chen and Patton, 1999), and (3) linear-parameter-varying (LPV) (Becker, 1993) and self-scheduling (Gahinet and Apkarian, 2013) approaches that allow obtaining directly a global controller. Despite numerous results, except for the recent structured H_∞ approach, the aeronautical industry has been conservative in adopting them. This reticence is partly due to a conservative design mindset (arising from the overriding safety concerns from carrying passengers), partly due academic research offering some times complex solutions demonstrated only in simple problems, and partly due to the lack of tools implementing the proposed methods. The latter was a significant concern for LPV methods until the recent appearance of mature modeling and synthesis tools, see (Balas et al., 2015b; Hjartarson et al., 2015). From the academic perspective, the issue has been often the difficulty in validating the techniques in real aircraft.

This article aims to present the validation of several H_∞ -based approaches (from standard to structured, and from individual to automated scheduled design) in a real aircraft, the Japan Aerospace Exploration Agency (JAXA) MuPAL-α (Sato and Satoh, 2011). Although the results are not from flight tests, they are performed in the aircraft onground in what is known as Aircraft-In-the-Loop (AIL) configuration, which is akin to the Iron-Bird configuration

⋆ This work was funded by the EU Horizon 2020 grant agreement No. 690 811 and Japan NEDO grant agreement No. 062800 for the project entitled "Validation of Integrated Safety-enhanced Intelligent flight cONtrol (VISION)". The first author also acknowledges funding for Beatriz Galindo Distinguished Researchers from the V-PRICIT framework of the Comunidad de Madrid with UC3M.

10.1016/j.ifacol.2022.11.291

used by Airbus (Goupil, 2011). For MuPAL-α, JAXA's flight team experience of over two decades indicate that the results from AIL validation are almost the same level as flight tests to assess the capability and performance of the controller. It is noted that it is unusual to find in the literature a comparison of such a breadth of techniques via their application to an actual complex system.

The description of JAXA's MuPAL-α aircraft, its configuration as an AIL validation test bench, the description of the linear models used for design (including transfer functions and numerical state-space values), as well as the Y* flight control architecture used as baseline are described in reference (Marcos et al., 2022) –and omitted here due to page restrictions. Thus, the next sections focus on: presenting the designs (Sec. 2), comparing their AIL validation results (Sec. 3), and the conclusions (Sec. 4).

2. H_∞ FTC DESIGNS

Ten designs are considered in this article, see Table 1, all developed and validated within the joint Europe-Japan VISION project, which had the aim of reducing pilot workload during abnormal conditions by increasing, through flight test validation, the technological readiness level (TRL) of smarter technologies for aircraft Guidance, Navigation and Control (GNC). A series of visits to the Chofu (Tokyo) branch of JAXA were undertaken by several European teams to flight and AIL test their respective designs. Each visit was for a duration of 2-weeks to allow for checking, verification, coding, and validation of the design(s) tested during the visit.

The design and validation programme followed by the authors of this article was to first obtain individual LTI designs using the structured (K_{rob}, and $K_{ftc-A,B}$) and standard ($K_{H_\infty-A,B,C}$) H_∞ approaches. Note that each of these two sets contains a robust, fault-passive controller (respectively, K_{rob} and $K_{H_\infty-A}$) and active, FTC designs. Then, their resulting global controllers ($K_{sched-1,2}$) were obtained by manually scheduling the corresponding set of 3 LTI controllers, using the same interpolation rule for both. Finally, automated scheduling approaches were also used: the so-called self-scheduled controller ($K_{selfsched}$) was founded on the design experience from the structured H_∞ LTI designs, while the LPV controller (K_{LPV}) was built up from the first 2 non-structured H_∞ designs.

For compactness of presentation the reader is referred to the specified references in the table for details on the synthesis of the structured H_∞ LTI designs (note that K_{ftc-B} is obtained as K_{ftc-A} but for different fault levels and airspeeds). The main thing to note here is that the synthesis of these controllers was increasingly complex, requiring only two plants for K_{rob} but 8 for K_{ftc-A} and K_{ftc-B}. More over, the latter required several iterations just for the selection of the fault combinations used for design. Initially, in order to try obtaining a single LTI controller valid for the full Loss of Efficiency (LoE) range, it was attempted to use all the combinations resulting from 0, 50 and 85% (18 plants in total since 2 V_{EAS} are considered). This proved too challenging for the optimization and yielded controllers with insufficient performance for the extreme cases (i.e. 0 and 85%). Further, it was observed that it was necessary to include the no-fault case even if the focus of K_{ftc-B} is the higher LoE range. Therefore, after several heuristic choices of potential combinations the set given in the table was used as a compromise between covering all the range but reducing the number of plants used for the multi-model structured-H_∞ synthesis. It would have been possible to simplify the design process and reduce the effort by focusing on designing the controllers at single LoE fault levels (as done for the standard H_∞ controllers, see next subsection). But the intention was to also demonstrate the multi-model capability of the structured-H_∞ approach and show the potential to use it to design simultaneously for several fault cases. Due to the specified structure, these LTI controllers all have only 2 states (arising from two pure integrators for the error signals, as in Fig. 2) and two static (feedforward K_{FF} and feedback K_{FB}) gain matrices that define the Y* law.

Similarly, for the manual-scheduled controllers $K_{sched-1,2}$, it is just necessary to note that each of the constituent LTI controllers operated individually for a specified LoE range or was linearly interpolated with its neighbor, and that both global controllers used the same interpolation rule/ranges. Specifically, and using $K_{sched-1}$ to exemplify the chosen rule: K_{rob} was applied for LoE<15%, K_{ftc-A} in [25-40]%, and K_{ftc-B} for LoE>50%, while in [15-25]% the first two are interpolated and in [40-50]% the last two. The interpolations are performed at the output of the controllers to avoid state mixture issues as well as allow for scheduling of controllers with different structures.

2.1 Active-fault tolerant, robust, standard LTI controllers

With respect to the full-order H_∞ designs, $K_{H_\infty-A,B,C}$, they were obtained in order to validate in MuPAL-α controllers synthesized with this more standard H_∞ approach, to compare the resulting designs with those obtained from the structured H_∞ approach, and also as a first step to subsequently design an FTC-LPV controller (since the LPV design technique used relies on a grid of standard H_∞

Table 1. Robust and FTC H_∞ designs (LoE := Loss of Efficiency faults).

1. K_{rob}	Non-FTC, multi-model structured-H_∞ controller robust to actuator uncertainty and airspeed changes. Two LTI models are used for the synthesis: at V_{EAS} 120 and 160 knots. Details in (Waitman et al., 2019).
2. K_{ftc-A}	FTC, multi-model structured-H_∞ design against aileron and rudder LoE faults. Eight LTI models are used for the synthesis: V_{EAS} at $\{120, 160\}$ knots and aileron/rudder LoE at $\{0, 40\}$%. Details in (Waitman et al., 2019).
3. K_{ftc-B}	FTC, as K_{ftc-A} but for V_{EAS} at $\{120, 130\}$ knots and ail/rud % LoE at: $\{0, 50\}$, $\{50, 0\}$, $\{50, 50\}$ and $\{85, 85\}$.
4. $K_{H_\infty-A,B,C}$	FTC, as the first 3 designs but using standard (i.e. non-structured) H_∞ synthesis. See Section 2.1.
5. $K_{sched-1,2}$	FTC, manual scheduling of respectively the first 3 designs (1, 2, 3) or the standard H_∞ ones ($K_{H_\infty-A,B,C}$). Both global controllers use the same interpolation scheme, a linear rule on the LoE fault level.
6. $K_{selfsched}$	FTC self-scheduled, structured-H_∞ design. See Section 2.2.
7. K_{LPV}	FTC linear parameter varying (LPV) design. Details in (Marcos et al., 2022).

-based designs). Each of these designs was obtained at a specific LoE level (0, 40, 80%) and at 120 knots airspeed.

The design interconnection for the full-order $K_{H_\infty-A,B,C}$ controllers is given in Fig. 1 where $\{G\#\}_{\#\in\{A,B,C\}}$ indicates the corresponding LTI plant model used and act_{LFT} the uncertain time delay actuators' model (in linear fractional transformation (LFT) format as detailed in (Sato and Satoh, 2011). The weights marked in blue shade ($W_{ref} = diag([5,1])\frac{\pi}{180}$ and $W_{\Delta_Z} = I_2$ with I_2 representing the unit matrix of dimension 2) are constant for the 3 controllers, while all the others vary in a coherent fashion as the LoE level increases {0, 40, 80}%, for example: $W_{\Delta_W} = \{1, 0.7, 0.5\}$ and $W_u = \{\frac{1}{5}I_2, \frac{1}{2.5}I_2, \frac{1}{1.5}I_2\}\ \frac{\pi}{180}$.

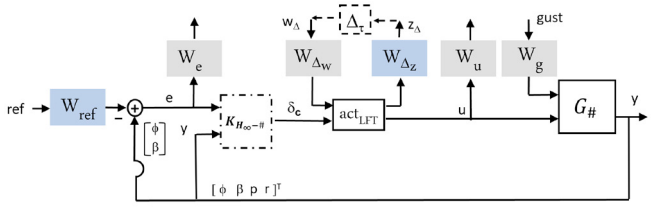

Fig. 1. MuPAL-α Y^* $K_{H_\infty-A,B,C}$ design interconnection.

As mentioned above the weights from these designs will be used as the basis for the design of K_{LPV}. The reader is referred to Table 2 in reference (Marcos et al., 2022) for the W_g and W_g weights at LoE {0, 40}%, plus explanations on their choice, while those at 80% are given by:

$$W_g = \frac{\frac{1}{50}s+5}{s+\frac{1}{10}}$$
$$W_e = \begin{bmatrix} \frac{s+1}{\frac{1}{0.04}s+0.01} & 0 \\ 0 & \frac{s+2}{\frac{1}{0.1}s+0.05} \end{bmatrix} \frac{180}{\pi} \quad (1)$$

The resulting LTI controllers have 11 states but prior to their discretization (see Sec. 2.4) their high-frequency modes are truncated down to 6 states.

2.2 Self-scheduling fault-tolerant, robust control

The self-scheduled FTC controller, $K_{selfsched}$, was designed at a fixed airspeed using another capability of the structured-H_∞ approach. As the first step, the scheduling parameter must be chosen and it was decided to use the overall aircraft fault level $\lambda = \max\{\gamma_a, \gamma_r\}$ (where γ_a and γ_r measure the active, independent aileron or rudder LoE fault). The choice of a single parameter simplifies the design process and implementation of $K_{selfsched}$, while the use of the maximum is for safety reasons as the worst case must always be considered, the drawback is that these choices may result in conservative results.

The next step, and more difficult, is to choose the scheduling function. A polynomial of order 2 was chosen for each of the two control gain matrices (K_{FF} and K_{FB}) that constitute the Y* law. The resulting scheduling function is: $K_\#(\lambda) = K_{\#,0} + K_{\#,1}\lambda + K_{\#,2}\lambda^2$, where $\# = \{_{FF,FB}\}$ and $\{K_{\#,i}\}_{i\in\{0,1,2\}}$ are the scheduling fitting matrices obtained from the synthesis. The fitting is performed for each element of the $K_{FF}(\lambda)$ and $K_{FB}(\lambda)$ gain matrices, resulting in 2×2 and 2×6 matrices of polynomials respectively. It helps to use K_{rob} and $K_{ftc-A,B}$ as the initial values of the optimization, by associating each to their corresponding fault level λ (i.e. {0, 0.35, 0.75}).

The synthesis was performed using the Matlab routine `systune` and the design interconnection of Fig.2 (where G_{act} and G are the actuator and plant LTI models, and K_{eff} is the actuator efficiency that represents its LoE fault level). A set of constraints on the sensitivity transfer function ($\|W_e(s)T_{r\to e}W_r\|_\infty < 1$), lateral-directional channels' cross-couplings ($\|W_{x,\phi_c\to\beta}T_{\phi_c\to\beta}(s)\|_\infty < 1$ and $\|W_{x,\beta_c\to\phi}T_{\beta_c\to\phi}(s)\|_\infty < 1$), high-frequency modes' damping ($\|W_rT_{r\to u}(s)W_u(s)\|_\infty < 1$), closed-loop poles ($\Re(p_i) < \sigma$), and overshoot (5%) were used and captured into a single block diagonal constraint.

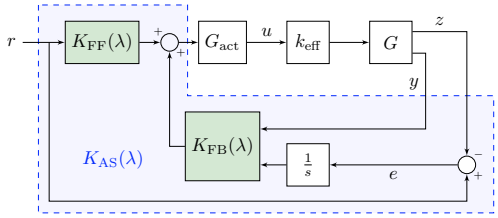

Fig. 2. MuPAL-α Y^* $K_{selfsched}$ closed-loop diagram.

2.3 LPV fault-tolerant, robust control

Finally, an FTC-LPV controller K_{LPV} was obtained using the Single-Quadratic-Lyapunov-Function (SQLF) method of (Becker, 1993). The details of this design are given in reference (Marcos et al., 2022) but just to mention here that: similar to the standard H_∞ approach the LPV method does not allow to specify the controller structure (only its input/outputs can be defined, see the design interconnection in Fig.3); it was obtained at a single airspeed (120 knots); it was based on the previous standard H_∞ designs (using the same actuator time delay uncertainty model and design weight structure but with different tuning and including LPV weights, marked in grey and with an *lpv* subscript); and, similar to the self-scheduled controller, it used the maximum of the $\{\gamma_a, \gamma_r\}$ faults as the single scheduling parameter.

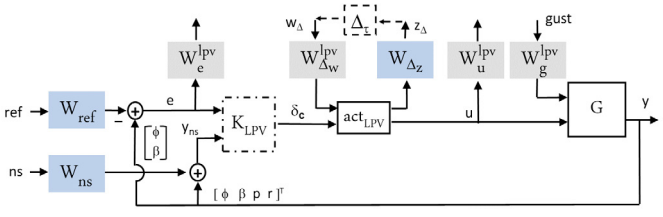

Fig. 3. MuPAL-α Y^* K_{LPV} design interconnection.

The LPV controller was synthesized using the *lpvsyn* function from *LPVTools* (Hjartarson et al., 2015) and a grid of two LoE fault levels, i.e. ρ_{LPV}=[0.0, 0.4]. It is important to note that for the design it is considered that both actuators are simultaneously affected. As it happened for the LTI design cases (but not for the self-scheduled design) it was not possible to obtain a single LPV controller with good performance from 0 up to 80% LoE faults. This was the case since the two extreme cases,

resulted in quite different dynamical behavior of the open loop and could not be handled simultaneously. It would have been possible to design another LPV controller for the higher fault region, but for ease of design and to demonstrate the flexibility of the implementation approach used, a different path was taken, see next.

2.4 Onboard implementation of controllers

For the implementation in the flight control computer, all the previous controllers need to be transformed from continuous to discrete time. This discretization is performed prior to the controllers' C-coding in the onboard Fly-By-Wire software. The sampling time of MuPAL-α's flight control computer is $T_s = 20$ ms and a Tustin transformation is used for the full-order standard-H_∞ and LPV controllers. For the structured-H_∞ designs, including $K_{selfsched}$, since the only dynamic element they contain is an integrator (see Fig.2), this block is simply discretized by replacing it with a numerical integrator with a trapezoidal rule as proposed by (Åström and Wittenmar, 2011).

In addition to the above, for K_{LPV} the implementation approach followed is that of (Marcos and Bennani, 2011), which extracts from the LPV controller at selected conditions a series of frozen-time LTI controllers (these are the ones discretized) and then uses a standard scheduling rule to interpolate their output signals. The LTI controllers thus extracted keep a high level of homogeneity among themselves as they all are derived from the same dynamical system, i.e. a single LPV controller, which is advantageous for the resulting global controller (Marcos and Balas, 2004). The main drawback is that this gridding extraction procedure implies the loss of the LPV synthesis' analytical guarantees, but it is typically the case that the resulting global controller is better than one formed from independently designed LTI controllers. An additional advantage of this gridding-extraction plus output-scheduling approach is that the implemented LPV controller can be augmented by including additional controllers. This was the case in here as the LPV controller obtained was designed only for a LoE region of [0-40]% so in order to cover the full range, the implemented LPV controller uses the two LTI controllers extracted from K_{LPV} (respectively at 0 and 40% LoE) augmented by the discretized version of $K_{H_\infty-C}$ (the standard-H_∞ design obtained at 80% LoE). The first two LTI-LPV controllers are implemented using the same ranges as for the manual-scheduled designs (i.e. LoE<15% and [15-25]%), but the interpolation between the 2^{nd} LTI-LPV controller and $K_{H_\infty-C}$ is performed between [60-80]% –beyond which only $K_{H_\infty-C}$ is active.

3. AIRCRAFT-IN-THE-LOOP VALIDATION

Each of the designs in Table 1 was analyzed and verified via linear frequency and time domain analysis (including Bode, gain/phase margins, step and doublet responses) for different airspeeds, faults, and actuator time delays prior to their AIL validation in MuPAL-α. During the latter, the designs underwent a plethora of AIL validation tests, which typically covered: individual and simultaneous doublets for the commands, gust/no-gust conditions, slow and fast airspeed configurations, and fault scenarios ranging from individual slow or fast LoE ramps to simultaneous LoE biases. In this section a subset of the AIL results is presented. It is noted that since the AIL validation campaign spanned several years it is not possible to present a single plot using the exact same scenario for all the designs (i.e. different magnitudes and time lengths were used for some designs in order to test better their capabilities). Nevertheless, from the linear verification and the AIL tests performed it is possible to state that the conclusions presented in this section apply across the designs.

Fig. 4 shows the results for a series of simultaneous δ_a and δ_r actuators' LoE ramp faults (see top row) and simultaneous bank and sideslip commands (black, solid lines in all the subsequent rows, with the left column for the bank response and the right for the sideslip). The fault ramps go from 0% to 80% LoE in a time length of 320 seconds (identical for both actuators). Starting from the 2^{nd} row and moving downwards, each row presents the responses for: (2^{nd} row) K_{rob}, (3^{rd} row) K_{ftc-A}, (4^{th} row) K_{ftc-B}, (5^{th} row) $K_{sched-1}$, (6^{th} row) $K_{sched-2}$, and (7^{th} row) $K_{selfsched}$. Results for the $K_{H_\infty-A,B,C}$ and K_{LPV} controllers are not included here for compactness. The latter is subsequently compared to $K_{selfsched}$ in Fig. 5, and with respect to the standard H_∞ design it suffices to look at their corresponding manual-scheduled design $K_{sched-2}$.

It is important to consider the clear improvement in response as we move down the list of controllers in order to understand the strengths and shortcomings of each controller. K_{rob} clearly shows a worsening of the decoupling design objectives as the faults' percentage augments. Even for small LoE levels, the sideslip response is always more affected than the bank angle (this is also due to the smaller magnitude of commands for β in comparison to those for ϕ, which also results in a more noticeable coupling from the latter on the former). Comparing this with the K_{ftc-A} and K_{ftc-B} active-FTC LTI controllers, it is seen that the first is able to maintain a similar response for LoE faults in the range $[0-40]$% but noticeably degrades at 80% (although still much better than K_{rob}). This indicates the improvement in response obtained by considering actively the effects of faults in the design process. On the other hand, K_{ftc-B} shows a degraded response than K_{ftc-A} on the [0-40]% LoE region (notice the much slower rise time and the coupling effects), but it is still acceptable (recall that the no-fault condition was included in the design), and as expected it shows a much improved fault-tolerance for LoEs $\in$ [40-80]% as it was designed for this region.

With respect to the scheduled designs: $K_{sched-1}$ shows that its responses are a combination of those from the individual controllers used to create it (the 3 previous ones), but with a slight improvement in the overall response (compare the [0-40] and [40-80]% LoE regions, but specially the mid time region [100-200] s, with those in the above three rows). Essentially, the manual-scheduled approach allows merging the individual designs and exploit their best capabilities at the most adequate regions to produce a more consistent overall performance. The assessment for $K_{sched-2}$, which uses the set of $K_{H_\infty-A,B,C}$ designs, is similar to that of $K_{sched-1}$: i.e. improved response across the LoE fault range. Comparing both manual-scheduled designs, it is seen that $K_{sched-2}$ shows some more ϕ command coupling in β from 100 seconds

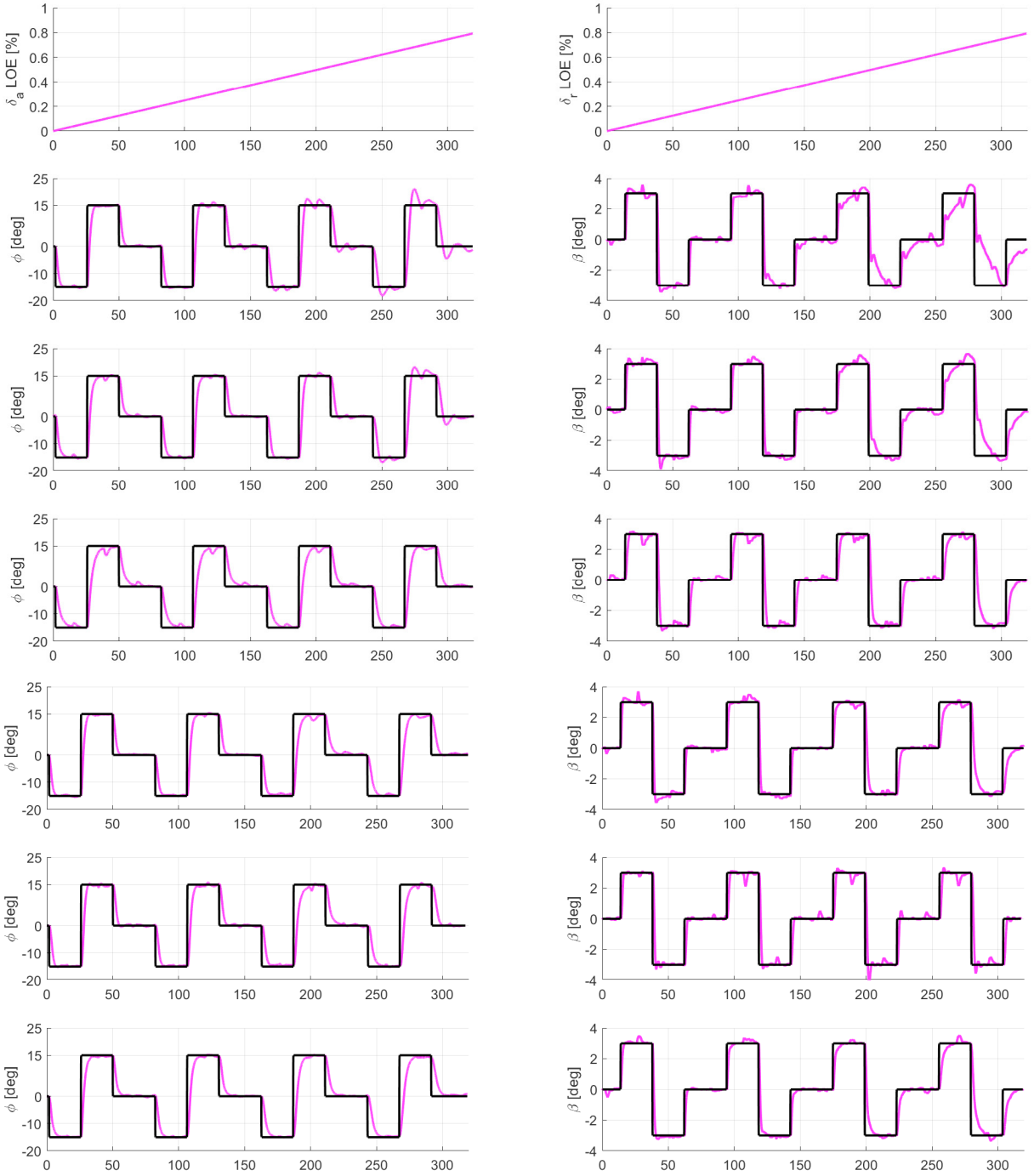

Fig. 4. JAXA MuPAL-α AIL validation: simultaneous LoE ramp faults (top row) and bank/sideslip commands for (from 2^{nd} row to bottom): K_{rob}, K_{ftc-A}, K_{ftc-B}, $K_{sched-1}$, $K_{sched-2}$, and $K_{selfsched}$.

onward in exchange for a faster rise and settling time in that region. It is concluded that despite the differences in the design process used between the two sets of LTI controllers, for the considered MuPAL-α FTC design case, the AIL validation results confirm that the resulting manually scheduled controllers have similar performance and robustness, which demonstrates the validity of the different methods and the advantages of the standard GS approach.

The last row of Fig. 4 shows that the self-scheduled design, $K_{selfsched}$, improves the responses with respect to that of the manual-scheduled controllers although it is more similar in performance to $K_{sched-1}$ as it was based on the structured-H_∞ LTI designs). It is critical to recognize that in contrast to the manual-scheduling designs, $K_{selfsched}$ provides design guarantees in-between points at the expense of a more complex design process.

The second AIL validation results shown, see Fig. 5, presents the results for the automated scheduled designs for the case where both actuators are affected by a bias fault of $-3°$(added in the aileron at $t = 50$ s and in the rudder at $t = 94$ s). Even though the controllers were not explicitly designed to be tolerant to such faults,

both $K_{selfsched}$ and K_{LPV} show very good fault tolerance behavior with very little effect from the faults. The shifted fault occurrence allows to clearly evaluate the no-fault performance in the first 50 seconds, which shows the type of rise times and very small overshoots obtained (which satisfy the control design objectives). Both controllers show similar responses for the ϕ command but a slight more coupling for K_{LPV}, while for the β command it is the opposite with the later controller showing a faster rise time and less coupling. These differences are mostly the result of the different design weights for each technique.

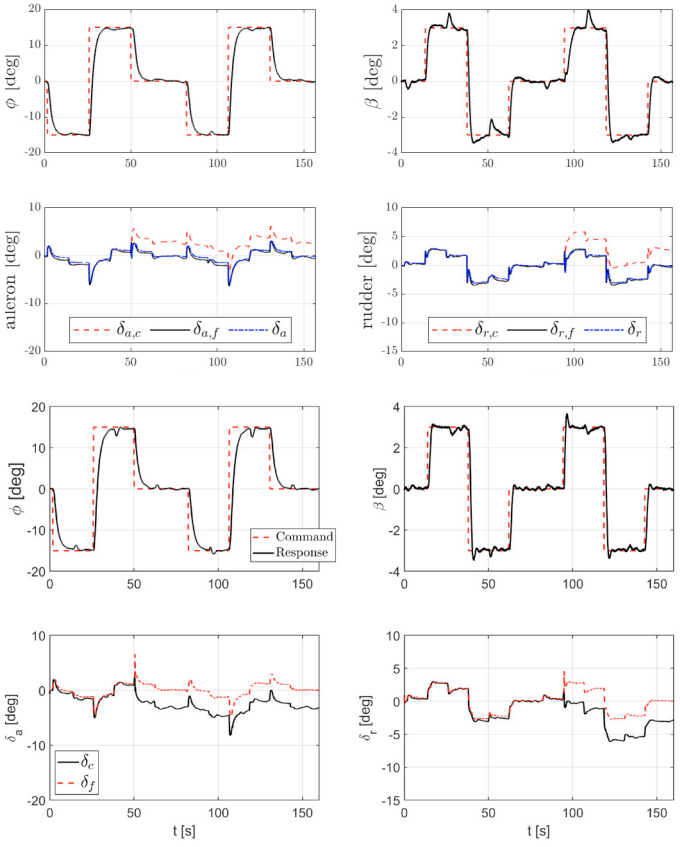

Fig. 5. MuPAL-α-AIL validation: δ_a and δ_r BIAS faults for $K_{selfsched}$ (top 2 rows) and K_{LPV} (bottom 2 rows)

4. CONCLUSION

A comparison of standard and structured LTI H_∞ designs, their manual-scheduled versions, and self-scheduled and LPV designs has been performed via Aircraft-In-the-Loop validations for two fault scenarios. All the designs proved successful, but the latter two showed more homogeneous performance and robustness across fault levels. All the approaches are methodological and rely on well-known and established theoretical developments. For the present case (i.e. aileron and rudder faults in MuPAL-α's lateral-directional motion), the LPV design was found to require less effort and time (although this design directly build on the standard H_∞ designs). Alll the scheduled designs were obtained and validated within a single visit of 2-weeks in JAXA (with two previous visits focused on the passive and active LTI controllers). This demonstrates the fast turn-around design process for all these techniques.

ACKNOWLEDGEMENTS

The authors thanks the support of JAXA's flight testing team, especially Mr. Hosoya Tomoaki, for all the help with the aircraft-in-the-loop validation activities.

REFERENCES

Åström, K. and Wittenmar, B. (1995). *Adaptive Control*. Addison-Wesley, New York, 2 edition.

Åström, K. and Wittenmar, B. (2011). *Computer-Controlled Systems: Theory and Design*. Dover Publications, 3rd edition.

Balas, G., Chiang, R., Packard, A., , and Safonov, M. (2015a). *Robust Control Toolbox: User's Guide*. The MathWorks.

Balas, G., Hjartarson, A., Packard, A., and Seiler, P. (2015b). LPVTools: A Toolbox for Modeling, Analysis, and Synthesis of Parameter Varying Control Systems, Software and User's Manual. Technical report, University of Minnesota, USA.

Becker, G. (1993). *Quadratic Stability and Performance of Linear Parameter Dependent Systems*. Ph.D. thesis, Department of Engineering, University of California, Berkeley.

Chen, L. and Patton, R. (1999). *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, The Netherlands.

Gahinet, P. and Apkarian, P. (2011). Structured H_∞ synthesis in MATLAB. *IFAC Proceedings Volumes*, 44(1), 1435–1440. 18th IFAC World Congress.

Gahinet, P. and Apkarian, P. (2013). Automated tuning of gain-scheduled control systems. 2740–2745. 52nd IEEE Conference on Decision and Control. doi: 10.1109/CDC.2013.6760297.

Goupil, P. (2011). Airbus state of the art and practices on FDI and FTC in flight control system. *Control Engineering Practice*, 19(6), 524–539.

Hjartarson, A., Packard, A., and Seiler, P. (2015). LPVTools: A Toolbox for Modeling, Analysis, and Synthesis of Parameter Varying Control Systems. In *1st IFAC Workshop on Linear Parameter Varying Systems*. Grenoble, Fr.

Marcos, A. and Balas, G. (2004). Development of Linear Parameter Varying Models for Aircraft. *AIAA Journal of Guidance, Control, and Dynamics*, 27(2), 218–228.

Marcos, A. and Bennani, S. (2011). A Linear Parameter Varying Controller for a Re-entry Vehicle Benchmark. In *1st Council of European Aerospace Societies (CEAS) GNC Conference*.

Marcos, A., Waitman, S., and Sato, M. (2022). Fault Tolerant Linear Parameter Varying Flight Control Design, Verification and Validation. *Journal of the Franklin Institute*, 359(2), 653–676.

Sato, M. and Satoh, A. (2011). Flight control experiment of Multipurpose-Aviation-Laboratory-α in-flight simulator. *Journal of Guidance, Control, and Dynamics*, 34(4), 1081–1096.

Waitman, S., Marcos, A., and Sato, M. (2019). Design and hardware-in-the-loop validation of a fault-tolerant Y* flight control law. In *4th International Conference on Control and Fault-Tolerant Systems (SYSTOL19)*. Casablanca, Morocco.