Automated Tuning of Gain-Scheduled Control Systems

Pascal Gahinet and Pierre Apkarian

Abstract— This paper describes an application of nonsmooth optimization to the tuning of fixed-structure gain-scheduled controllers. The gain schedule is tuned over the entire operating range subject to standard frequency-domain requirements at each design point. The result is a smooth, compact scheduling formula that can be either implemented directly or turned into a lookup table. The effectiveness of this approach is illustrated on a gain-scheduled three-loop autopilot for the pitch axis of an airframe.

I. INTRODUCTION

Gain scheduling is a well established practice in industry with many applications in automotive, aerospace, and industrial automation. It is commonly used to control nonlinear systems or linear systems whose dynamics change with time and/or operating condition. Typically, gain-scheduled controllers are fixed single- or multi-loop control structures where lookup tables specify gain values as a function of the scheduling variables. The conventional approach consists of dividing the system's operating range into regions where linear control is adequate, designing a linear controller for each region, and "interpolating" the resulting set of controllers in some reasonable way (gain interpolation or control signal blending). This ad-hoc strategy yields satisfactory results as long as the linearized dynamics vary slowly during operation [1]. Alternative approaches based on linear matrix inequalities (LMI) have been proposed in [2]-[6]. While they provide stability and performance guarantees, they come with various degrees of conservatism and also dictate the controller structure and complexity. See [7], [8] for a comprehensive survey of the state of the art.

loop autopilot of Figure 1 for controlling the flight path and vertical acceleration of an airframe. Because the aerodynamic coefficients change significantly with speed V and angle of attack α , the autopilot gains are typically scheduled as a function of (α, V) . In practice, these gains are tuned either manually or via optimization at each flight condition and are then globally adjusted to avoid sudden changes and provide smooth transition between flight conditions. In [9], the authors use nonsmooth optimization to tune the gains at each flight condition and discuss how to build a gain schedule from these results. In this paper, we push this idea one step further and show how nonsmooth optimization can be used to directly tune the entire gain schedule without need for post-processing or interpolation. This results in a smooth gain schedule with consistent performance at all design points. Moreover, the compact formula expressing the dependence on the scheduling variables provides a memoryfriendly alternative to a large lookup table.

The paper is organized as follows. Section 2 describes the proposed approach in the context of Linear Parameter-Varying (LPV) systems. Section 3 reviews the rationale for using nonsmooth optimization when tuning fixed-structure controllers and discusses extensions to the tuning of gainscheduled controllers. Finally, Section 4 demonstrates the effectiveness of our approach on a realistic aerospace application.



Fig. 1. Three-loop autopilot.

II. PROBLEM SETUP

This paper is concerned with conventional gain scheduling for fixed control structures. A classic example is the threeOur starting point is any control system where the plant is nonlinear or time-varying, the feedback structure is predefined, and the control elements are linear compensators whose parameters must be adjusted as a function of some scheduling variables. The autopilot of Figure 1 is one such example, but what follows is by no means limited to this specific architecture. Since any linear compensator can be

P. Gahinet is with The MathWorks, 3 Apple Hill, Natick, MA 01760-2098, USA. Pascal.Gahinet@mathworks.com

Pierre Apkarian is with ONERA, 2, av. Ed. Belin, 31055, Toulouse, France. Pierre.Apkarian@onera.fr

written as an interconnection of elementary gain and integrator blocks, we assume without loss of generality that all gain-scheduled elements are (SISO or MIMO) static gains. Using simple block diagram manipulations, we can transform any control structure into the *Standard Form* of Figure 2 where

- σ denotes the vector of scheduling variables. These variables can include physical parameters and system inputs, outputs, or states.
- K₁(σ),...,K_N(σ) are the controller gains to be tuned and scheduled as a function of σ
- \mathcal{P} is the nonlinear model obtained by combining the plant with everything but the tunable gains $K_1(\sigma), \ldots, K_N(\sigma)$
- w and z are signals of interest for expressing the control objectives (see Section 3 for details).



Fig. 2. Standard form for fixed-structure gain scheduling.

In this form, the gain scheduling problem consists of finding functions $K_1(\sigma), \ldots, K_N(\sigma)$ that achieve the performance goals for the closed-loop system of Figure 2. This remains a daunting task given that \mathcal{P} is a nonlinear system and $K_1(\sigma), \ldots, K_N(\sigma)$ are unknown functions of σ . Following the conventional gain scheduling approach, we then approximate \mathcal{P} by a Linear Parameter-Varying (LPV) model $P(s, \sigma)$ of the form

$$\dot{x} = A(\sigma)x + B_1(\sigma)w + B_2(\sigma)u \tag{1}$$

$$z = C_1(\sigma)x + D_{11}(\sigma)w + D_{12}(\sigma)u$$
 (2)

$$y = C_2(\sigma)x + D_{21}(\sigma)w + D_{22}(\sigma)u.$$
 (3)

This family of linear models is usually obtained by linearizing the plant dynamics at operating conditions $(x(\sigma), w(\sigma), u(\sigma))$ parameterized by σ [7]. For example, the airframe model in Figure 1) is linearized at trim conditions $\dot{\alpha} = \dot{q} = 0$ for a range of incidence and speed values $(\sigma = (\alpha, V))$.

Conventional gain scheduling proceeds by (a) selecting a finite number of design points $\sigma_1, \ldots, \sigma_M$ covering the operating range, (b) tuning the gains K_1, \ldots, K_N at each design point for the linearized plant $P(s, \sigma_m)$, and (c) smoothing the resulting set of gains across design points. Note that this

process does not require explicit knowledge of the functions $A(\sigma), B_1(\sigma), \ldots$ Instead we just need their values at the design points $\sigma_1, \ldots, \sigma_M$, which can be obtained by M linearizations of \mathcal{P} .

To eliminate the need for a-posteriori smoothing and obtain a compact scheduling formula, we propose using a finite expansion of $K_i(\sigma)$ of the form

$$K_j(\sigma) = K_{j0} + f_{j1}(\sigma)K_{j1} + \ldots + f_{jL}(\sigma)K_{jL}$$
 (4)

where $f_{j1}(.), \ldots, f_{jL}(.)$ are user-selected "basis" functions and the coefficients K_{j0}, \ldots, K_{jL} are the tunable parameters. There are many possible choices of basis functions, from generic choices such as polynomials to problem-specific choices guided by engineering insight and past experience. Using this expansion and replacing \mathcal{P} by the linearized model $P(s, \sigma_m)$, we can rearrange the closed-loop system of Figure 2 to look like Figure 3 at the design point σ_m , where the linear model $P_m(s)$ depends on $P(s, \sigma_m)$ and the basis function values $f_{jl}(\sigma_m)$.



Fig. 3. Linearized standard form at design point σ_m .

Summing up, by selecting a discrete set of design points in the operating range, linearizing the plant dynamics at each design point, and using a finite expansion of the tunable gains, we have reduced the original gain scheduling problem to the following multi-model "robust" tuning problem:

(GST) Find K_{jl} values that ensure adequate closed-loop performance in Figure 3 for all plant models $P_1(s), \ldots, P_M(s)$.

Note that the coefficients K_{jl} are tuned against the entire set of design points. We call this approach *Gain Surface Tuning* (GST) since we are effectively shaping entire gain surfaces. As shown next, this problem is tractable with available nonsmooth optimization tools when using standard frequency-domain performance criteria.

III. OPTIMIZATION-BASED TUNING

In principle, we could choose any number of performance criteria and optimization algorithms to quantify and solve the GST problem. In practice, tractability and effectiveness vary widely with these choices. Here we propose using the nonsmooth techniques of [10], [11]. While this may not be the best choice in all situations, this approach has proven very effective in solving similar fixed-structure tuning problems [12]–[16]. There are two main components to this approach:

- 1) Performance is quantified using standard frequencydomain metrics such as the H_{∞} norm (peak gain across frequency), the H_2 norm (average output power for white noise inputs), and the damping and natural frequency of the closed-loop poles. Different criteria applying to different transfer functions can be mixed together. For example, we can specify the bandwidth and loop gain of a feedback loop, limit the sensitivity and control effort, enforce adequate stability margins, and regulate stochastic disturbances all at the same time.
- 2) The resulting multi-objective, multi-model optimization program is solved with dedicated optimizers capable of overcoming the nonsmooth nature of the H_{∞} norm and of competing objectives.

Specifically, let x denote the vector of optimization variables obtained by collecting all tunable entries in the coefficients K_{jl} of (4). At a given design point σ_m , the performance objectives are of the form (ignoring spectral constraints for simplicity):

$$\|W_L(s)T_m(s,x)W_R(s)\| \tag{5}$$

where $T_m(s, x)$ denotes the closed-loop transfer function from w to z in Figure 3, W_L and W_R combine frequency weighting and I/O channel selection, and $\|.\|$ stands for the H_2 or H_∞ norm. Collectively, the GST problem is therefore equivalent to the program:

$$\min_{x} \max_{i,m} f_{im}(x) \quad \text{subject to} \quad \max_{j,m} g_{jm}(x) \le 1 \qquad (6)$$

where the functions $f_{im}(x)$ and $g_{jm}(x)$ are of the form (5) and the indices *i* and *j* are relative to the set of requirements. We refer to the terms $f_{im}(x)$ as the soft requirements (objectives to be minimized) and to the terms $g_{jm}(x)$ as the hard requirements (constraints to be enforced). Note that non-smoothness stems primarily from the fact that the H_{∞} norm is a maximum over frequency:

$$||H(s,x)||_{\infty} = \max \sigma_{\max}(H(j\omega,x))$$

As we try to push down the peak gain, other peaks arise (waterbed effect) and we soon end up with multiple active peaks. Jointly pushing them down then requires finding a suitable descent direction in the subdifferential of the $\max_{\omega} \sigma_{\max}(.)$ function.

The nonsmooth program (6) can be solved with an appropriate extension of the algorithm described in [10] and implemented in the systume function of [16]. Details of the algorithm are beyond the scope of this paper. This algorithm is guaranteed to find critical points (local minima) of (6). While this is a non-convex program, a few runs with randomized initial conditions are usually sufficient to weed out undesirable local minima. By working with the Standard Form of Figure 3 and exploiting the nature of the

performance criteria (5), this algorithm is amenable to highperformance implementations.

A suitable solution x of (6) determines the coefficients K_{jl} for the gain surface (4). As with the convential approach, the resulting gain-scheduled controller comes with no global guarantees of performance, so validation on a finer σ grid and nonlinear simulations remain necessary to fully qualify the results.

IV. EXAMPLE

This section applies the "Gain Surface Tuning" method described in Sections 2 and 3 to the three-loop autopilot of Figure 1. A schematic of the airframe appears in Figure 4. The autopilot must track a command γ_{ref} in flight path angle by controlling the normal acceleration A_z and the pitch rate q. PI control is used for the pitch rate loop and static gains are used for the acceleration and flight path loops. Because the aerodynamic forces and moments vary with the incidence angle α and the speed V, the autopilot gains must be scheduled as a function of α and V. For the operating range considered here, α varies between -20 and +20 degrees and V varies between 700 to 1400 m/s. We use an airframe model similar to the one described in [9], [17]. The 3-dof equations for the longitudinal motion are

$$\dot{u} = -g\sin\theta + qw + (F_x(\alpha, V, \delta) + T)/m \quad (7)$$

$$\dot{v} = -g\cos\theta - qu + F_z(\alpha, V, \delta)/m$$
 (8)

1

$$\dot{q} = M(\alpha, V, \delta) / I_{yy} \tag{9}$$

$$\dot{\theta} = q \tag{10}$$

where g is the gravitational acceleration, $\theta = \gamma + \alpha$ is the pitch angle, F_x, F_z, M are the aerodynamic forces and moments, T is the thrust, and m and I_{yy} are the mass and moment of inertia of the airframe. The actuator is modeled as a second-order system with natural frequency $\omega_n = 150$ rad/s and damping $\zeta = 0.7$. When neglecting gravity, the airframe equations are symmetric in α and it is therefore enough to consider positive values of α or, equivalently, to use $|\alpha|$ as scheduling variable.



Fig. 4. Airframe nomenclature.

For tuning purposes, we select a 5-by-9 grid of linearly spaced design points with α ranging from 0 to 20 degrees and V from 700 to 1400 m/s. To derive an LPV model of the form (1)-(3), we linearize (7)-(10) at the 45 design points for the trim condition $\dot{w} = \dot{q} = 0$ (zero normal acceleration

and pitching moment). The gain surfaces (4) are chosen to be first-order polynomial in α and V:

$$K_p(\alpha, V) = K_{p0} + \alpha \times K_{p1} + V \times K_{p2} + \alpha V \times K_{p3}$$
(11)

and similarly for the other gains K_i, K_a, K_g . Finally, the autopilot is tuned to meet the following requirements:

- The flight path angle γ should track the reference $\gamma_{\rm ref}$ with a response time of about 0.5 seconds. In terms of the transfer function S_g from γ to the tracking error e_g , this is expressed as $||W_g S_g||_{\infty} < 1$ where the weighting function $W_g(s) = 0.77(s+2)/(s+0.03)$ emphasizes the frequency range from 0 to 1 rad/s.
- The A_z loop should track well at low frequency and should roll off past 10 rad/s. This is expressed as $||W_a S_a||_{\infty} < 1$ where S_a is the closed-loop transfer from e_a to A_z and $W_a(s) = 0.06(s+1.5)(s+13.7)/(s+0.025)$.
- The autopilot should reject disturbances at the plant input. This is quantified as $||W_d T_d||_{\infty} < 1$ where T_d is the closed-loop transfer from δ to A_z and $W_d(s) = (0.25s + 1)/(150s)$ emphasizes rejection at low frequency.
- The closed-loop poles should have a minimum damping of 0.35. This is expressed as $\psi(p) := \max_p 2 + \operatorname{Re}(p)/(0.35|p|) < 1$ where p are the closed-loop poles.

The resulting nonsmooth program (6) has objective function

$$f(x) = \max(\|W_g S_g\|_{\infty}, \|W_a S_a\|_{\infty}, \|W_d T_d\|_{\infty}, \psi(p))$$
(12)

and no g(x) constraint. For comparison purposes, we performed three separate designs:

- 1) Classic Gain Scheduling: Independently tune one set of gains K_p , K_i , K_a , K_g at each design point (α, V)
- 2) Gain Surface Tuning: Tune the 16 coefficients of the gain surfaces (11) for $K_p(\alpha, V), ..., K_g(\alpha, V)$ to meet the requirements at *all* design points
- 3) **Robust Design**: Tune a single set of gains K_p, K_i, K_a, K_g to meet the requirements at *all* design points

The first design establishes a baseline for the best achievable performance at each design point. The optimized value for f(x) was found to range between 0.99 and 1.11 (a value less than 1 indicates that all requirements are met). The GST approach achieved a best overall value of 1.13 and produced the following gain surfaces (in terms of the normalized scheduling variables $\bar{\alpha} = (\alpha - 0.1745)/0.349$ and $\bar{V} = (V - 1050)/350$):

$$\begin{split} K_p(\alpha, V) &= 0.1(1 + 0.071 \,\bar{\alpha} + 0.068 \,\bar{V} - 0.048 \,\bar{\alpha}\bar{V}) \\ K_i(\alpha, V) &= 3.6 - 0.054 \,\bar{\alpha} + 0.39 \,\bar{V} - 0.27 \,\bar{\alpha}\bar{V} \\ K_a(\alpha, V) &= 0.001(5.7 - 0.52 \,\bar{\alpha} - 3.3 \,\bar{V} + 0.076 \,\bar{\alpha}\bar{V}) \\ K_g(\alpha, V) &= -2700 - 340 \,\bar{\alpha} - 900 \,\bar{V} - 180 \,\bar{\alpha}\bar{V} \end{split}$$

Finally, the robust design achieved a best overall value of 1.48, significantly worse than the first two which confirms the benefits of gain scheduling for this application. The three designs were performed with systume [16] with

computation times of 36, 27, and 20 seconds, respectively (using a 3GHz Intel Xeon processor with 12 GB of RAM).



Fig. 5. Tracking of step change in flight path angle.



Fig. 6. Tracking of acceleration command in A_z loop.



Fig. 7. Rejection of disturbance at plant input.

Figure 5-7 compare the tracking and disturbance rejection performance for the three designs. Note that Gain Surface Tuning yields homogenous responses free of "outliers" and on par with the pointwise optima. It also yields smooth gain surfaces, which is not the case when the gains are tuned independently at each design point as shown in Figure 8. Finally, we implemented the gain schedules $K_p(\alpha, V), \ldots$ from the second design and simulated the controller's ability to execute a maneuver that takes the airframe through a large portion of its operating range. The simulation results are summarized in Figure 9 and confirm good tracking performance despite rapid changes in speed and incidence. For more details on the model and results, see related example in [16].



Fig. 8. K_p surfaces for classic and GST approaches.





Fig. 9. Nonlinear simulation of "Gain Surface Tuning" design.

Note: In the nonlinear simulation, our gain-scheduled controller does not make use of the offsets associated with the trim condition $\dot{w} = \dot{q} = 0$. This is justified by the fact that for the model considered here, the short-period dynamics are essentially described by a quasi-LPV model. To see this, observe that V varies slowly compared to α and q so $\dot{w} \approx -u\dot{\alpha}$. Ignoring gravity and the slow dynamics along u and using $\cos \alpha \approx 1$, (7)-(10) simplify to

$$\dot{\alpha} = q - \frac{F_z(\alpha, V, \delta)}{mV}, \quad \dot{q} = \frac{M(\alpha, V, \delta)}{I_{uu}}.$$

Using the particular structure of the aerodynamic coefficients [17], this can be rewritten as

$$\dot{\alpha} = q - \frac{F_{\alpha}(\alpha, V)}{mV}\alpha - \frac{F_{\delta}(\alpha, V)}{mV}\delta$$
(13)

$$\dot{q} = \frac{M_{\alpha}(\alpha, V)}{I_{yy}}\alpha + \frac{M_{\delta}(\alpha, V)}{I_{yy}}\delta$$
(14)

which is a quasi-LPV model of the short-period dynamics with state (α, q) , input δ , and varying parameter $\sigma = (\alpha, V)$. Consequently, we can ignore the trim offsets and act as if we were controlling (13)-(14).

V. CONCLUSION

We have presented a novel way to tune gain-scheduled controllers. Our approach closely follows the conventional gain-scheduling workflow except that it seeks to tune the entire gain schedule at once. Nonsmooth optimization tools are used to optimize the gain surfaces, which enables us to (1) tune fixed-structure control systems with pre-defined feedback structure and control elements, and (2) use a variety of well-established frequency-domain criteria to express the control objectives. The validity and effectiveness of this approach have been illustrated on a realistic three-loop autopilot application.

REFERENCES

- J. F. Shamma and M. Athans, "Guaranteed Properties of Gain Scheduled Control for Linear Parameter-varying Plants," *Automatica*, vol. 27, no. 3, pp. 559–564, march 1991.
- [2] G. Becker and A. Packard, "Gain-Scheduled State-Feedback with Quadratic Stability for Uncertain Systems," in *Proc. American Control Conf.*, Boston, MA, 1991, pp. 2702–2703.
- [3] A. Packard, "Gain scheduling via linear fractional transformations," Syst. Control Letters, vol. 22, pp. 79–92, 1994.
- [4] P. Apkarian and P. Gahinet, [™]A convex characterization of gainscheduled H_∞ controllers," *IEEE Trans. Aut. Control*, vol. 40, no. 5, pp. 853–864, May 1995, see also pp. 1681.
- [5] C. Scherer, "Robust Generalized H₂ Control for Uncertain and LPV Systems with General Scalings," in *Proc. IEEE Conf. on Decision and Control*, Kobe, JP, 1996, pp. 3970–3975.
- [6] C. Scherer and E. Kose, "Gain-scheduled control synthesis using dynamic d-scales," *IEEE Trans. Aut. Control*, vol. 57, pp. 2219–2234, 2012.
- [7] W. Rugh and J. F. Shamma, "Research on Gain Scheduling," Automatica, vol. 36, pp. 1401–1425, 2000.
- [8] J. Mohammadpour and C. Scherer, *Control of Linear Parameter Varying Systems with Applications*. Springer, 2012.
- [9] L. Ravanbod and D. Noll, "Gain-Scheduled Two-Loop Autopilot for an Aircraft," *7th IFAC Symposium on Robust Control Design*, pp. 772– 777, 2012.
- [10] P. Apkarian and D. Noll, "Nonsmooth H_{∞} synthesis," *IEEE Trans.* Aut. Control, vol. 51, no. 1, pp. 71–86, 2006.
- [11] J. V. Burke, D. Henrion, A. S. Lewis, and M. L. Overton, "HIFOO a MATLAB package for fixed-order controller design and H_{∞} optimization," in *5th IFAC Symposium on Robust Control Design*, Toulouse, France, July 2006.
- [12] P. Gahinet and P. Apkarian, "Decentralized and fixed-structure H_{∞} control in MATLAB," in *Proc. CDC-ECC*, Orlando, Florida, Dec 2011, pp. 8205–8210.
- [13] S. Hirwa, P. Feyel, G. Duc, and G. Sandou, "Reduced-order controllers for inertial line of sight stabilization," in *Proc. 7th IFAC Symposium* on Robust Control Design, 2012, pp. 103–108.
- [14] N. Guy, D. Alazard, C. Cumer, and C. Charbonnel, "Reduced-order H_{∞} controller synthesis for flexible structures control," in *Proc. 7th IFAC Symposium on Robust Control Design*, 2012, pp. 109–114.
- [15] G. Puyou and P. Ezerzere, "Tolerance of aircraft longitudinal control to the loss of scheduling information: Toward a performance-oriented approach," in *Proc. 7th IFAC Symposium on Robust Control Design*, 2012, pp. 393–399.
- [16] Robust Control Toolbox 5.0. MathWorks, Natick, MA, USA, Sept 2013.
- [17] S. Bennani, D. Willemsen, and C. Scherer, "Robust control of linear parametrically varying systems with bounded rates," J. Guidance, Control, and Dynamics, vol. 21, pp. 916–922, 1998.