

# Robustness via structured $H_\infty/H_\infty$ synthesis

Laleh Hosseini-Ravanbod <sup>\*</sup>    Dominikus Noll <sup>\*</sup>    Pierre Apkarian <sup>†</sup>

## Abstract

Multi-objective  $H_\infty/H_\infty$ -synthesis with structured control laws is discussed and used as a means to enhance robustness of the system in the presence of real parametric uncertainty.

*Keywords:* Structured  $H_\infty$  synthesis, multi-objective  $H_\infty/H_\infty$  synthesis, structured control law, robustness versus performance.

## 1 Introduction

In multi-objective control design several closed-loop performance specifications have to be optimized simultaneously. A specific form is multi-objective  $H_\infty/H_\infty$ -synthesis, where the  $H_\infty$  norm of a first closed loop performance channel  $\|T_{w \rightarrow z}(K)\|_\infty$  is minimized subject to a constraint  $\|T_{\tilde{w} \rightarrow \tilde{z}}(K)\|_\infty \leq \gamma$  on a second one. Here we discuss *structured  $H_\infty/H_\infty$  synthesis*, where the controller is in addition constrained to have a specific pattern. Controller structures or *pattern* are dictated by practical considerations and include reduced and low-order controllers, decentralized or PID controllers, observer-based controllers, or even control architectures where simple controllers are combined with feed-forward, set-point or washout filters, and much else.

In this work we use structured  $H_\infty/H_\infty$ -synthesis to achieve a trade-off between performance and robustness specifications. We select a channel  $w \rightarrow z$  to describe good performances of the system, and a second channel  $\tilde{w} \rightarrow \tilde{z}$  to assess robustness. Then  $H_\infty/H_\infty$ -synthesis is a practical way to optimize performance subject to a constraint assuring a satisfactory level of robustness.

It is well-known that input or output sensitivity functions can, when measured in the  $H_\infty$ -norm, give robustness certificates with respect to unstructured system uncertainty [33]. Here the system is *desensitized*, but the structure of the uncertainty is not taken into account. As soon as one is interested in the more specific case of real parametric uncertainty, the structured singular value  $\mu$  according to Doyle *et al.* [33] is a mathematically rigorous way to assess robustness. Unfortunately,  $\mu$  is NP-hard even to evaluate in most cases, [10, 11], which means that it is not fit to be used within an optimization procedure, where functions and constraints are evaluated many times. Here we show that the distance to instability can offer a compromise between an accurate representation of the parametric uncertainty, and a convenient use within an optimization procedure.

The structure of the paper is as follows. Section 2 recalls the concept of controller structure and structured synthesis. Parametric uncertainty is briefly reviewed in section 3. Semi-structured

---

<sup>\*</sup>Université Paul Sabatier, Institut de Mathématiques, Toulouse, France

<sup>†</sup>ONERA, Department of Systems Control and Flight Dynamics, Toulouse, France

stability radii are discussed in section 4, and the trade-off between performance and robustness is presented subsequently. In section 5 we use contour plots to analyze the degree of conservatism in using stability radii. Section 6 presents methods to compute structured mixed  $H_\infty/H_\infty$  controllers. Section 7 gives the link with  $\mu$  and its approximation  $\tilde{\mu}$ , which can be used for robustness analysis of a controller computed with the help of a stability radius. Regulation of the roll axis of a flexible geostationary satellite is discussed in the experimental section 8.

## 2 Structured synthesis

Following [3] a feedback controller in state-space form

$$(1) \quad K : \begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix}$$

with  $A_K$  of size  $n_K \times n_K$ ,  $B_K$  of size  $n_K \times p_2$ ,  $C_K$  of size  $m_2 \times n_K$  and  $D_K$  of size  $m_2 \times p_2$  is called *structured* if the state space matrices depend smoothly on a design parameter  $\theta$  varying in some parameter space  $\mathbb{R}^n$  or some constrained subset of  $\mathbb{R}^n$ :

$$(2) \quad A_K = A_K(\theta), B_K = B_K(\theta), C_K = C_K(\theta), D_K = D_K(\theta).$$

Writing  $K = K(\theta)$  for short, this notion is particularly useful if the dimension  $n$  of  $\theta$  is smaller than the total number of degrees of freedom  $N := n_K^2 + n_K m_2 + n_K p_2 + m_2 p_2$  in (1). Controller structures arise naturally in practical situations and include for instance reduced and low-order controllers, PID, decentralized, observer-based controllers, or controller architectures combining simple controllers with feed forward, setpoint or washout filters.

Two typical examples can be seen in (3). PID controllers are parametrized as  $K_{\text{pid}}(\theta)$ , where  $\theta = (\tau, \text{vec}(R_i), \text{vec}(R_d), \text{vec}(D_K)) \in \mathbb{R}^{3m_2 p_2 + 1}$ , while observer-based controllers  $K_{\text{obs}}(\theta)$  have  $\theta = (\text{vec}(K_c), \text{vec}(K_f)) \in \mathbb{R}^{n_x m_2 + n_x p_2}$ .

$$(3) \quad K_{\text{pid}}(\theta) = \begin{bmatrix} 0 & 0 & R_i \\ 0 & -\tau I & R_d \\ I & I & D_K \end{bmatrix}, \quad K_{\text{obs}}(\theta) = \left[ \begin{array}{cc|c} A - B_2 K_c - K_f C_2 & K_f \\ \hline -K_c & 0 \end{array} \right].$$

Structured  $H_\infty$ -synthesis is now as follows. Given an open loop plant

$$(4) \quad P : \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}$$

with state  $x \in \mathbb{R}^{n_x}$ , measured output  $y \in \mathbb{R}^{m_2}$ , controlled input  $u \in \mathbb{R}^{p_2}$ , external inputs  $w$  and regulated outputs  $z$ , and fixing a controller structure (2), we have to compute an output feedback controller with that pre-defined structure  $K(\theta^*)$  in (2) which stabilizes  $P$  internally and minimizes the  $H_\infty$  norm of the closed-loop performance channel  $T_{w \rightarrow z}(K(\theta))$  among all stabilizing controllers of the same structure  $K(\theta)$ . For short, we compute a solution  $\theta^*$  of the optimization program

$$(5) \quad \begin{array}{ll} \text{minimize} & \mathcal{P}(\theta) = \|T_{w \rightarrow z}(K(\theta))\|_\infty \\ \text{subject to} & K(\theta) \text{ closed-loop stabilizing} \end{array}$$

We refer to  $\mathcal{P}(\theta)$  as the performance specification or simply as the performance in closed-loop, with  $\theta$  indicating that controller  $K(\theta)$  is used.

**Remark 1.** The standard  $H_\infty$  synthesis problem is a special case of (5) if  $K(\theta)$  stands for the class of full order controllers (1), where full-order means  $n_K = n_x$  and that all gains  $A_K, B_K, \dots$  are free optimization variables. In this case  $n = N$ , which is why we sometimes refer to full order controllers en abus de langue as *unstructured*. It is well-known [18] that the standard  $H_\infty$ -problem is equivalent to a convex problem, which can be solved via AREs or LMIs, available via the Matlab functions `HINFRIC` or `HINFLMI`. Imposing the structural constraint  $K = K(\theta)$  makes (5) non-convex as a rule. Even for very simple structures like fixed reduced order controllers, where  $n_K < n_x$ , computation of a global optimum in (5) is NP-hard (see [10,11]). Computing the global minimum is then out of the question and we accept locally optimal solutions.

The structured  $H_\infty/H_\infty$ -problem can now be introduced in a similar way. Given the plant

$$(6) \quad \tilde{P} : \begin{bmatrix} \dot{x} \\ \tilde{z} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & \tilde{B}_0 & B_1 & B_2 \\ \tilde{C}_0 & \tilde{D}_{00} & \tilde{D}_{01} & \tilde{D}_{02} \\ C_1 & \tilde{D}_{10} & D_{11} & D_{12} \\ C_2 & \tilde{D}_{20} & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ \tilde{w} \\ w \\ u \end{bmatrix}$$

and a controller structure (2), we have to find the optimal solution  $\theta^*$  of the following constrained optimization program.

$$(7) \quad \begin{aligned} & \text{minimize} && \mathcal{P}(\theta) = \|T_{w \rightarrow z}(K(\theta))\|_\infty \\ & \text{subject to} && \mathcal{R}(\theta) = \|T_{\tilde{w} \rightarrow \tilde{z}}(K(\theta))\|_\infty \leq \gamma \\ & && K(\theta) \text{ closed-loop stabilizing} \end{aligned}$$

Regarding the fact that (7) is non-convex as a rule, we accept locally optimal solutions. For reasons which will become clear shortly, we refer to  $\tilde{w} \rightarrow \tilde{z}$  as the robustness channel, and to  $\mathcal{R}$  as the robustness criterion or simply as the robustness.

### 3 Parametric robustness

In parametric robustness the state-space representation (4) depends rationally on a set of real uncertain parameters  $\delta_1, \dots, \delta_r$ , which are arranged in block-diagonal matrices of the form

$$(8) \quad \Delta = \begin{bmatrix} \delta_1 I_{m_1} & & & \\ & \ddots & & \\ & & \delta_r I_{m_r} & \end{bmatrix},$$

where  $m_i \in \mathbb{N}$ ,  $m = m_1 + \dots + m_r$ . The set of matrices  $\Delta$  with structure (8) is denoted  $\mathbf{\Delta}$ , and we refer to  $\Delta \in \mathbf{\Delta}$  as a real parametric uncertainty, where  $\delta_i$  is called repeated if  $m_i > 1$  and scalar if  $m_i = 1$ . Via normalization the set of uncertainties over which we wish to guarantee robustness is usually of the form

$$(9) \quad Q = \{\Delta \in \mathbf{\Delta} : \sigma_1(\Delta) \leq 1\},$$

where  $\sigma_1$  denotes the maximum singular value. The robust design problem is now the following. Given the family of plants

$$(10) \quad P(\Delta) : \begin{bmatrix} \dot{x} \\ p \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_0 & B_1 & B_2 \\ C_0 & D_{00} & D_{01} & D_{02} \\ C_1 & D_{10} & D_{11} & D_{12} \\ C_2 & D_{20} & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x \\ q \\ w \\ u \end{bmatrix}, \quad q = \Delta p$$

indexed by the uncertainty  $\Delta \in \mathbf{\Delta}$ , we wish to compute an output feedback controller  $u = Ky$  such that

1.  $K = K(\theta^*)$  has the pre-defined structure (2) and stabilizes  $P(\Delta)$  internally in closed loop for every uncertainty  $\Delta \in Q$ .
2.  $K(\theta^*)$  minimizes the closed-loop performance objective  $\mathcal{P}(K(\theta))$  among all other controllers  $K(\theta)$  with the same structure satisfying item 1.

Arranging (10) in such a way that the controller  $K = K(\theta)$  is static, closing the loop with the controller and leaving the loop with  $\Delta$  in (10) open shows that  $K$  stabilizes each  $P(\Delta)$  internally if and only if each matrix

$$(11) \quad A(K, \Delta) = A + B_2KC_2 + (B_0 + B_2KD_{20}) \Delta (I - D_{00}\Delta - D_{02}KD_{20}\Delta)^{-1} (C_0 + D_{02}KC_2)$$

is Hurwitz. We may therefore present the following idealized optimization program

$$(12) \quad \begin{aligned} & \text{minimize} && \mathcal{P}(K) = \|T_{w \rightarrow z}(K)\|_\infty \\ & \text{subject to} && A(K, \Delta) \text{ Hurwitz for every } \Delta \in Q \\ & && K = K(\theta) \end{aligned}$$

If we define the structured stability radius as

$$(13) \quad r_{\mathbf{\Delta}}(A, B, C, D) = \inf\{\sigma_1(\Delta) : \Delta \in \mathbf{\Delta}, A + B\Delta(I - D\Delta)^{-1}C \text{ unstable}\},$$

then as a consequence of (9) program (12) may be cast more accurately as

$$(14) \quad \begin{aligned} & \text{minimize} && \mathcal{P}(K) \\ & \text{subject to} && \mathcal{R}(K) = r_{\mathbf{\Delta}}(A(K), B(K), C(K), D(K))^{-1} \leq 1 - \epsilon \\ & && K = K(\theta) \end{aligned}$$

where  $\epsilon > 0$  is some small threshold. Here we have written the matrix  $A(K, \Delta)$  in (11) as

$$(15) \quad A(K, \Delta) = A(K) + B(K)\Delta(I - D(K)\Delta)^{-1}C(K)$$

with  $A(K) = A + B_2KC_2$ ,  $B(K) = B_0 + B_2KD_{20}$ ,  $C(K) = C_0 + D_{02}KC_2$ ,  $D(K) = D_{00} + D_{02}KD_{20}$ . Notice that  $r_{\mathbf{\Delta}}^{-1}$  coincides with the structured singular value  $\mu_{\mathbf{\Delta}}$  of [33] in the special case where only real uncertain parameters are present.

**Remark 2.** Notice that our problem differs from the robust control problem of Zhou [33] mainly by the fact that we allow arbitrary controller structures  $K(\theta)$ , while classical  $\mu$ -theory works only with full order (unstructured) controllers  $K$ . We could also consider robust performance, in which case the nominal performance objective  $\mathcal{P}(K)$  would have to be replaced by  $\mathcal{P}(K) = \sup_{\Delta \in Q} \|T_{w \rightarrow z}(P(\Delta), K)\|_\infty$ .

**Remark 3.** The difficulty in (12) or (14) can be gauged by the fact that computation of  $\mu_{\mathbf{\Delta}}$ , and therefore also  $r_{\mathbf{\Delta}}$ , is NP-hard (in the state-space dimension of  $\mathcal{F}_\ell(P, K)$ ), so that solving program (14) frontally is out of the question. Realistically, we may just be able to perform robustness analysis of a given controller over  $Q$  at reasonable cost, i.e., check whether  $r_{\mathbf{\Delta}}(\mathcal{F}_\ell(P, K)) > 1$ .

## 4 Semi-structured stability radius

As the structured stability radius  $r_{\Delta}$  is generally inaccessible, we propose to measure robustness via relaxed forms of the stability radius. Formula (15) suggests to the following semi-structured stability radius [29]:

$$(16) \quad r_{\mathbb{F}}(A, B, C, D) = \inf\{\sigma_1(\Delta) : A + B\Delta(I - D\Delta)^{-1}C \text{ unstable, or } \det(I - D\Delta) = 0, \Delta_{\nu\mu} \in \mathbb{F}\},$$

where  $\mathbb{F} = \mathbb{R}$  or  $\mathbb{F} = \mathbb{C}$  and  $\Delta_{\mu\nu}$  are the elements of matrix  $\Delta$ . Clearly  $r_{\mathbb{F}} \leq r_{\Delta}$ , so that  $r_{\mathbb{F}}(A(K), B(K), C(K), D(K)) > 1$  implies robust stability of  $A(K)$  over  $Q$ . We expect  $r_{\mathbb{F}}$  to be conservative in the sense that  $r_{\mathbb{F}} \ll r_{\Delta}$ , because in (16) we replace structured  $\Delta \in \Delta$  by an arbitrary matrix  $\Delta$  of the same size. The conservatism will be addressed in section 5. We now consider the following relaxation of (12):

$$(17) \quad \begin{aligned} & \text{minimize} && \mathcal{P}(K) = \|T_{w \rightarrow z}(K)\|_{\infty} \\ & \text{subject to} && \mathcal{R}(K) = r_{\mathbb{F}}(A(K), B(K), C(K), D(K))^{-1} \leq r^{-1} \\ & && K = K(\theta) \end{aligned}$$

The difficulty here is to choose the parameter  $r$ . The choice  $r = 1$  would provide a robustness certificate, but we expect it to be too conservative, so that we would either fail to satisfy the constraint, or if we succeed, would spoil performance. Several ideas to calibrate  $r$  can be employed. In generally we use the following scheme:

### Algorithm I

---

Input:  $Q$ , structure  $K(\theta)$ , plant  $P$ . Output: Solution  $K(\theta^*)$  of (17) robustly stable over  $Q$ .

---

- 1: **Nominal synthesis.** Compute optimal solution  $\theta_1$  to (5). Its performance is  $p_1 = \mathcal{P}(\theta_1)$ . Check parametric robustness using  $\mu_{\Delta}$  respectively  $\tilde{\mu}_{\Delta}$ . If  $K(\theta_1)$  is robustly stable over  $Q$  quit, otherwise continue. Compute  $r_{\mathbb{F}}(A(K(\theta_1)), B(K(\theta_1)), C(K(\theta_1)), D(K(\theta_1))) =: r_1 > 0$ .
- 2: **Calibrate.** Compute solution  $\theta_2$  of

$$(18) \quad \min_{\theta \in \mathbb{R}^n} r_{\mathbb{F}}(A(K(\theta)), B(K(\theta)), C(K(\theta)), D(K(\theta)))^{-1}$$

and let  $r_2^{-1}$  be the value of (18).

- 3: **Constrained synthesis.** Given  $r$  between  $r_1$  and  $r_2$ , solve constrained program (17). The solution is  $K(\theta^r)$ .
  - 4: **Robustness analysis.** Use  $\mu_{\Delta}$  or its approximation  $\tilde{\mu}_{\Delta}$  to analyze parametric robustness of  $K(\theta^r)$  over  $Q$ . If it holds stop and return  $\theta^* = \theta_r$ . If  $K(\theta^r)$  is too robust (hence lacking in performance), choose larger  $r \in (r_1, r_2)$ , if  $K(\theta^r)$  lacks robustness choose smaller  $r \in (r_1, r_2)$ . Then go back to step 3.
- 

In the above scheme we have the option to replace program (17) by a dual version in which the roles of constraint and objective are interchanged. More precisely, let  $\theta_1$  be the solution of the nominal program (5) obtained in step 1, with  $p_1 = \mathcal{P}(\theta_1)$  denoting nominal performance. If  $K(\theta_1)$  fails to be parametrically robust over  $Q$ , the distance to instability of  $A(K(\theta_1))$  is likely to

be too small, and in consequence we expect performance to be too good, (i.e.  $p_1$  too small). In step 3 of algorithm I we could then consider the following dual version of (17):

$$(19) \quad \begin{aligned} & \text{minimize} && \mathcal{R}(\theta) = r_{\mathbb{F}}(A(K(\theta)), B(K(\theta)), C(K(\theta)), D(K(\theta)))^{-1} \\ & \text{subject to} && \mathcal{P}(\theta) \leq (1 + \alpha)p_1 \\ & && K(\theta) \text{ closed-loop stabilizing} \end{aligned}$$

Here we accept a loss of  $100\alpha\%$  over nominal performance  $p_1$  and use this freedom to increase the semi-structured distance  $r_{\mathbb{F}}$  to instability. The question is now how to choose  $\alpha$ . The lower bound being  $\underline{\alpha} = 0$ , we may use the solution  $\theta_2$  of (18) to get the upper bound  $\bar{\alpha} = \mathcal{P}(\theta_2)/p_1 - 1$ . The rest of the algorithm I now goes without change.

Computation of  $r_{\mathbb{F}}$  has been investigated by many authors. In the complex case we have  $r_{\mathbb{C}}(A, B, C, D)^{-1} = \|C(sI - A)^{-1}B + D\|_{\infty}$  as long as  $A$  is Hurwitz. This can be seen e.g. from Hinrichsen and Pritchard [21]. As a consequence, for  $\mathcal{P} = r_{\mathbb{C}}^{-1}$  programs (17) and (19) are structured  $H_{\infty}/H_{\infty}$  synthesis problems and therefore specific instances of (7). For that we have to augment the plant  $P$  into a plant  $\tilde{P}$  according to (6) such that  $\|C(K)(sI - A(K))^{-1}B(K) + D(K)\|_{\infty}$  becomes the  $H_{\infty}$  norm of the channel  $\tilde{w} \rightarrow \tilde{z}$ .

The real case is more complicated to analyze. Qiu *et al.* [29] prove that for  $A$  stable and  $G(s) = C(sI - A)^{-1}B + D$ ,

$$r_{\mathbb{R}}^{-1}(A, B, C, D) = \sup_{\omega \in [0, \infty]} \inf_{0 < \gamma \leq 1} \sigma_2 \left( \begin{bmatrix} \text{Re } G(j\omega) & -\gamma \text{Im } G(j\omega) \\ \gamma^{-1} \text{Im } G(j\omega) & \text{Re } G(j\omega) \end{bmatrix} \right),$$

where  $\sigma_2$  is the second largest singular value of a matrix. While function values of  $r_{\mathbb{R}}$  have been computed by Tits *et al.* [30], subgradient information is more difficult to obtain, so that making  $\mathcal{R} = r_{\mathbb{R}}^{-1}$  fit for the numerical approach (17) is more complicated. In the following section we investigate whether the reduction of conservatism obtained by using  $r_{\mathbb{R}}$  instead of  $r_{\mathbb{C}}$  justifies the additional work.

## 5 Analysis of conservatism

In order to decide how conservative the constraints  $r_{\mathbb{F}}^{-1} \leq r^{-1}$  in (17) are, we look at polar plots, or as they are sometimes called, spectral value sets. For a given controller  $K$  we are interested in the set

$$(20) \quad \Lambda(K) = \{\lambda \in \mathbb{C} : \lambda \text{ eigenvalue of } A(K, \Delta) \text{ for some } \Delta \in Q\}$$

of all closed loop poles of the uncertain matrices  $A(K, \Delta)$ ,  $\Delta \in Q$ . Namely,  $K$  is robustly stabilizing over  $Q$  as soon as  $\Lambda(K) \subset \mathbb{C}^-$ , where  $\mathbb{C}^-$  is the open left half plane. The link with  $r_{\Delta}$  in (13) is the following. If  $q = \max\{\sigma_1(\Delta) : \Delta \in Q\}$ , then  $\Lambda(K) \subset \mathbb{C}^-$  iff  $r_{\Delta}(K)^{-1} < q^{-1}$ . Satisfying the constraint in (12) with  $r = q$  by optimizing  $K = K(\theta)$  is therefore the same as pushing  $\Lambda(K)$  into the left half plane  $\mathbb{C}^-$ . As  $r_{\Delta}$  is NP-hard to compute, so is  $\Lambda(K)$ . For a single visualization with few uncertainties this may be possible, but in general we need to consider approximations.

A first approximation which comes to mind uses the *pseudo-spectrum*, a classical tool to analyze properties of non-normal matrices and operators, see e.g. Trefethen [31]. Writing  $A(K, \Delta) = A(K) + E(K, \Delta)$ , one may replace the structured perturbation  $E(K, \Delta)$  by an unstructured  $E$  of the same size with appropriate norm bound. Putting

$$e(K) = \sup\{\sigma_1(E(K, \Delta)) : \Delta \in Q\}$$

we define

$$(21) \quad \Lambda_{1,\mathbb{F}}(K) = \{\lambda \in \mathbb{C} : \lambda \text{ eigenvalue of } A(K) + E \text{ for some } E \\ \text{with } \sigma_1(E) \leq e(K) \text{ and } E_{\mu\nu} \in \mathbb{F}\}.$$

Clearly  $\Lambda(K) \subset \Lambda_{1,\mathbb{R}}(K) \subset \Lambda_{1,\mathbb{C}}(K)$ , so that  $\Lambda_{1,\mathbb{F}}(K) \subset \mathbb{C}^-$  implies robust stability, but at the cost of some conservatism. Namely, as  $\Lambda(K)$  might be a small subset of  $\Lambda_{1,\mathbb{F}}(K)$ , placing  $\Lambda_{1,\mathbb{F}}(K)$  in the left half plane  $\mathbb{C}^-$  may mean overdoing it and pushing  $\Lambda(K) \subset \mathbb{C}^-$  too far to the left, leading to a severe loss of performance. This conservatism can now be visualized via the gap between the contours  $\Lambda(K)$  and  $\Lambda_{1,\mathbb{F}}(K)$  in the polar plot (see Figure 1). Notice that contours  $\Lambda_{1,\mathbb{F}}$  correspond to using  $r_{\mathbb{F}}(A, I, I, 0)$  in program (7) or (19). This distance to instability is sometimes noted  $\beta(A)$  and related to the spectral abscissa  $\alpha(A)$  of a matrix. Stability optimization based on  $\alpha(A)$  has been proposed in [14]. Our graphical evaluation in Figure 8 shows that pseudo-spectra are very conservative so that the use of  $\beta(A(K))$  in (17) should in our opinion not be encouraged. Moreover, the fact that  $e(K)$  depends on  $K$  shows that in principle  $r$  should also depend on  $K$ , which is not manageable.

In order to reduce conservatism, we go back to (11) and write  $A(K, \Delta)$  as

$$(22) \quad A(K, \Delta) = A(K) + B(K)F(K, \Delta)C(K),$$

with  $B(K) = B_0 + B_2KD_{20}$ ,  $C(K) = C_0 + D_{02}KC_2$ , and  $F(K, \Delta) = \Delta(I - D_{00}\Delta - D_{02}KD_{20}\Delta)^{-1}$ . Putting

$$f(K) = \sup\{\sigma_1(F(K, \Delta)) : \Delta \in Q\}$$

we consider the following polar sets with semi-structured perturbations of the form

$$(23) \quad \Lambda_{2,\mathbb{F}}(K) = \{\lambda \in \mathbb{C} : \lambda \text{ eigenvalue of } A(K) + B(K)FC(K) \text{ for some } F \\ \text{with } \sigma_1(F) \leq f(K) \text{ and } F_{\mu\nu} \in \mathbb{F}\}.$$

By construction we have  $\Lambda \subset \Lambda_{2,\mathbb{R}} \subset \Lambda_{2,\mathbb{C}}$ , so that  $\Lambda_{2,\mathbb{F}}(K) \subset \mathbb{C}^-$  is sufficient for robust stability. With the use of  $\Lambda_{2,\mathbb{F}}$  we expect conservatism to be somewhat reduced, even though we cannot be sure that  $\Lambda_{2,\mathbb{F}} \subset \Lambda_{1,\mathbb{F}}$ . Notice that there is a fair chance that  $D_{02} = 0$  or  $D_{20} = 0$ , in which case  $F(\Delta) = F(K, \Delta)$  no longer depends on  $K$ , so that  $f = f(K)$  is a constant. This is indeed what happens in our numerical example studied in section 8. The contours  $\Lambda_{2,\mathbb{F}}$  correspond to using  $r_{\mathbb{F}}(A, B, C, 0)$  in programs (7), (17) or (19), see section 8. This type of contour has been discussed in [20] under the term *spectral value sets*. In our experiment  $\Lambda_{2,\mathbb{C}}$  gives much better results than  $\Lambda_{1,\mathbb{C}}$  (see Figure 9), but this may no longer be the case if  $f = f(K)$  depends on  $K$ , as then  $r$  in (17) should in principle also depend on  $K$ , respectively,  $\theta$ .

In order to avoid dependence of  $r$  on  $K$ , we decompose  $F(K, \Delta)$  in (22) as

$$F(K, \Delta) = \Delta(I - D(K)\Delta)^{-1},$$

where  $D(K) = D_{00} + D_{02}KD_{20}$ . Letting

$$q = \sup\{\sigma_1(\Delta) : \Delta \in Q\},$$

the following set of poles makes sense:

$$(24) \quad \Lambda_{3,\mathbb{F}}(K) = \{\lambda \in \mathbb{C} : \lambda \text{ eigenvalue of } A(K) + B(K)\Delta(I - D(K)\Delta)^{-1}C(K) \\ \text{for some } \Delta \text{ with } \sigma_1(\Delta) \leq q, \Delta_{\mu\nu} \in \mathbb{F}\}.$$

We have  $\Lambda \subset \Lambda_{3,\mathbb{R}} \subset \Lambda_{3,\mathbb{C}}$ , so that  $\Lambda_{3,\mathbb{F}}(K) \subset \mathbb{C}^-$  implies robust stability of  $K$  over  $Q$ . Moreover, we expect  $\Lambda_{3,\mathbb{F}}$  to be even less conservative than  $\Lambda_{2,\mathbb{F}}$ , although there is no inclusion between the two contours. Contours  $\Lambda_{3,\mathbb{F}}$  correspond to using the full  $r_{\mathbb{F}}(A, B, C, D)$  in the optimization programs. Figure 10 confirms that the third type of contours produces the best results.

Sets of the form  $\Lambda_{i,\mathbb{F}}(K)$  are generally referred as pseudo-spectra of  $A(K)$ .  $\Lambda_{1,\mathbb{R}}, \Lambda_{1,\mathbb{C}}$  are known as the complex and real pseudo-spectrum, while  $\Lambda_{2,\mathbb{F}}, \Lambda_{3,\mathbb{F}}$  are sometimes called structured (real or complex) pseudo-spectra. Here we prefer the nomination *semi-structured pseudo spectrum*, because the truly structured pseudo-spectrum is  $\Lambda$  in (20).

**Proposition 1.** *Let  $f_{3,\mathbb{C}}(s) = 1/\sigma_{\max}(G(s))$ , where  $G(s) = C(sI - A)^{-1}B + D$ . Then  $\Lambda_{3,\mathbb{C}} = \{s \in \mathbb{C} : f_{3,\mathbb{C}}(s) \leq q\}$ .*

This is indeed Theorem 5.2.16 in [23]. The functions  $f_{1,\mathbb{C}}$  and  $f_{2,\mathbb{C}}$  defining the contours  $\Lambda_{1,\mathbb{C}}$  and  $\Lambda_{2,\mathbb{C}}$  are obtained as special cases. For  $\Lambda_{2,\mathbb{C}}$  one puts  $D = 0$  (and  $\Delta$  becomes  $F$ ,  $q$  is replaced by  $f$ ). For  $\Lambda_{1,\mathbb{C}}$  one puts  $D = 0, B = I, C = I$  (and  $F$  becomes  $E$ ,  $f$  is replaced by  $e$ ).

Computing real pseudo-spectra is slightly more involved. Defining  $f_{3,\mathbb{R}} : \mathbb{C} \rightarrow \mathbb{R}$  as

$$f_{3,\mathbb{R}}(s) = \left( \inf_{0 < \gamma \leq 1} \sigma_2 \begin{bmatrix} \operatorname{Re} G(s) & -\gamma \operatorname{Im} G(s) \\ \gamma^{-1} \operatorname{Im} G(s) & \operatorname{Re} G(s) \end{bmatrix} \right)^{-1},$$

where  $G(s) = C(sI - A)^{-1}B + D$ , we have the following result, which is essentially based on [29] and can be found as Theorem 5.2.31 in [23].

**Proposition 2.**  $\Lambda_{3,\mathbb{R}} = \{\lambda \in \mathbb{C} : f_{3,\mathbb{R}}(\lambda) \leq q\}$ .

Computing pseudo-spectra  $\Lambda_{1,\mathbb{C}}$  via contour plots of  $f_{1,\mathbb{C}}$  is discussed at length in Trefethen [31], for  $\Lambda_{1,\mathbb{R}}$  in Trefethen and Embree [32]. The argument in proposition 1 is essentially based on Hinrichsen and Pritchard [21], while proposition 2 uses the argument of Qiu *et al.* [29]. Karow *et al.* [24] give further information.

Based on propositions 1 and 2, we compute

$$(25) \quad \Lambda_{3,\mathbb{C}}^\rho = \{z \in \mathbb{C} : \log_{10}(\max(\operatorname{svd}(C(zI - A)^{-1}B + D))) \leq \rho\},$$

and similarly for  $\Lambda_{3,\mathbb{R}}^\rho$ . (Here  $\rho = \log_{10} q$  for the  $q$  in Proposition 2). We discretize  $z = x + jy$  on a  $250 \times 250$  grid.

There are two aspects which we hope to learn from the polar plots. Firstly, as already mentioned, inspecting the gap between  $\Lambda$  and the  $\Lambda_{i,\mathbb{F}}$  allows to gauge the degree of conservatism, which is considerable. Secondly, we want to compare  $\Lambda_{i,\mathbb{R}}$  to  $\Lambda_{i,\mathbb{C}}$  in order to understand whether  $r_{\mathbb{R}}^{-1}$  should be preferred to  $r_{\mathbb{C}}^{-1}$  in (7). Notice that  $r_{\mathbb{C}}$  is easier to handle, so that the reduction in conservatism needed to justify the use of  $r_{\mathbb{R}}$  has to be significant.

**Remark 4.** Concerning the question  $r_{\mathbb{R}}$  versus  $r_{\mathbb{C}}$ , we can see immediately that  $f_{i,\mathbb{R}}(x) = f_{i,\mathbb{C}}(x)$  for real  $x$ . This is confirmed by checking the point on the real axis where the complex and the real contour meet (Figures 1, 3, 5). As soon as this point is the rightmost point of the complex contour, we cannot expect  $r_{\mathbb{R}}$  to offer any advantage over  $r_{\mathbb{C}}$ .

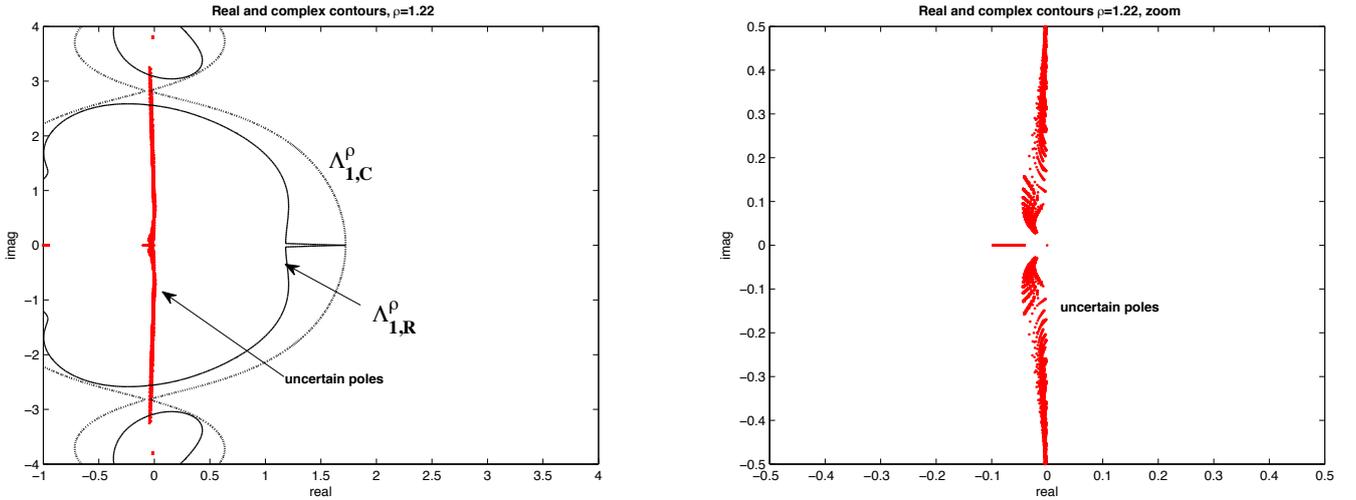


FIGURE 1: Contours  $\Lambda_{1,\mathbb{R}}(K)$ ,  $\Lambda_{1,\mathbb{C}}(K)$  at level  $\rho = 1.22$  on the left (with  $\rho = \log_{10} e$  for  $e$  as in (21)). The real contour shows the typical pointed-beak on the right. The beak touches the complex contour at the rightmost point on the real axis. Right image shows zoom on  $\Lambda(K)$ .

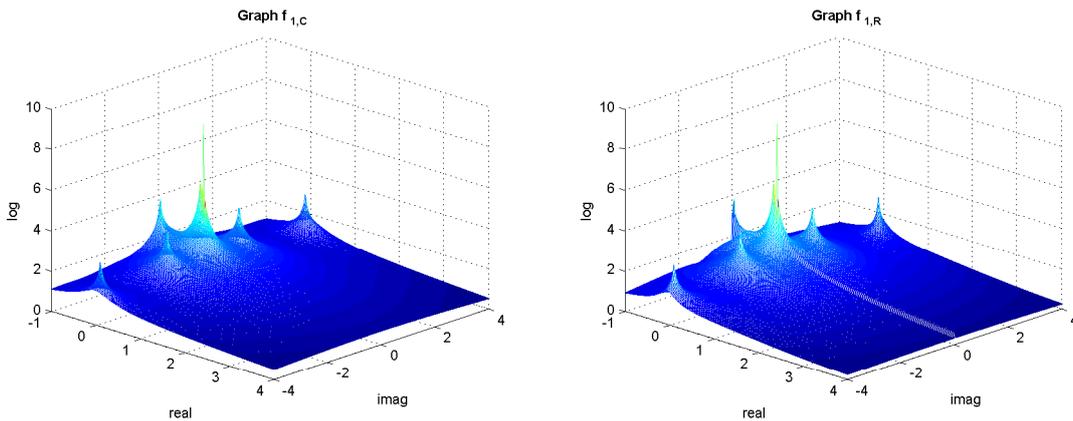


FIGURE 2: Graphs of  $f_{1,\mathbb{C}}$  on the left,  $f_{1,\mathbb{R}}$  on the right.

## 6 Solving the $H_\infty/H_\infty$ program

In this section we consider numerical strategies to solve the structured  $H_\infty/H_\infty$  program. The first numerical approach we are aware of is [5], where this problem has been treated as a *multidisk problem*. The multidisk problem was introduced and motivated in [25], and the novelty in [5] over e.g. [25] is the presence of the structural constraint  $K = K(\theta)$ .

More recently, in [17], mixed  $H_\infty/H_\infty$  is used to synthesize a structured controller in longitudinal flight control of a civil aircraft. Here  $K(\theta)$  stands for an ensemble of PID controllers and pre-filters. The authors use a progress function method which was previously analyzed in [6, 7] for mixed  $H_2/H_\infty$ -synthesis, and can be traced back to [28]. In [2] we have extended this to  $H_\infty$ -synthesis subject to time domain constraints. The progress function method may be slow, as it tries to achieve feasibility  $\mathcal{R} \leq \gamma$  in (7) respectively  $\mathcal{R} \leq r^{-1}$  in (17) before the objective starts getting reduced.

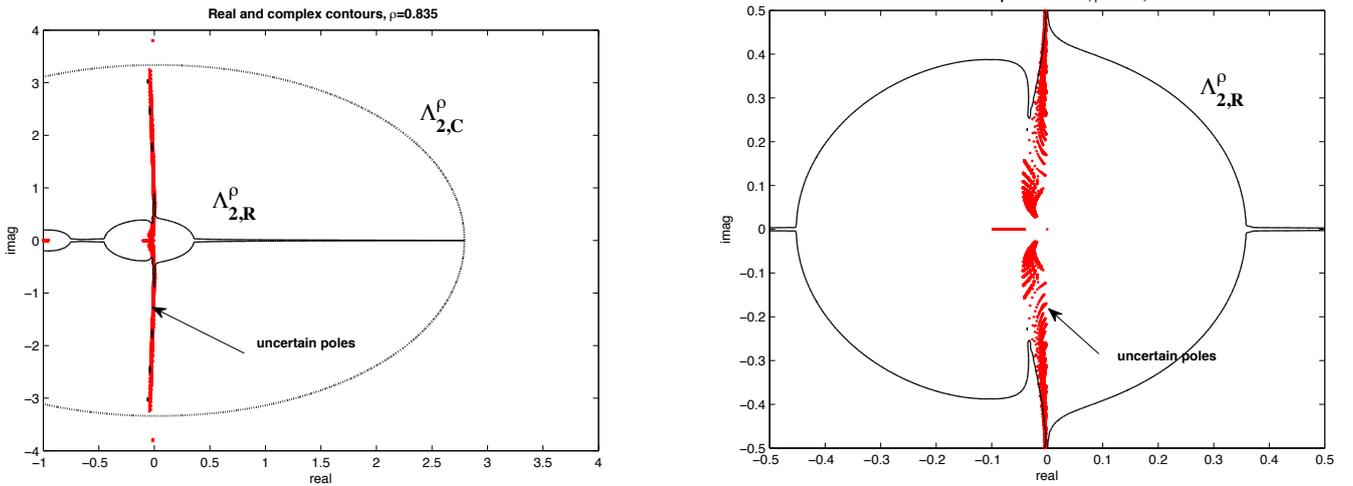


FIGURE 3: Contours  $\Lambda_{2,\mathbb{R}}(K)$  (continuous) and  $\Lambda_{2,\mathbb{C}}(K)$  (dashed-dotted) at  $\rho = 0.835$  on the left, zoom on a neighborhood of the origin on the right (where  $\rho = \log_{10} f$  with  $f$  as in (23)). The real contour is globally much closer to  $\Lambda(K)$  than the complex, but the extremely pointed beak spoils this advantage.

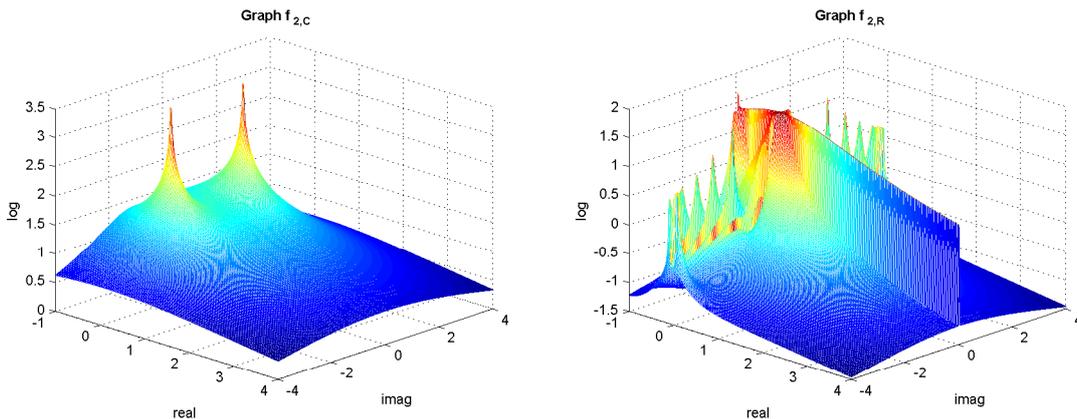


FIGURE 4: Graphs of  $f_{2,\mathbb{C}}$  left and  $f_{2,\mathbb{R}}$  right. The function defining the real contour is very spiky.

An alternative way to address (17) uses a succession of structured  $H_\infty$  problems. Indeed, locally program (7) is equivalent to the following unconstrained structured  $H_\infty$ -program

$$(26) \quad \begin{aligned} & \text{minimize} && \max\{\|T_{w \rightarrow z}(K(\theta))\|_\infty, \beta \|T_{\tilde{w} \rightarrow \tilde{z}}(K(\theta))\|_\infty\} \\ & \text{subject to} && K(\theta) \text{ closed-loop stabilizing} \end{aligned}$$

which for fixed  $\beta > 0$  is a specific form of (5) if a suitable weighting is introduced in (4) and the performance channel is  $(w, \tilde{w}) \rightarrow (z, \tilde{z})$ . We have the following

**Proposition 3.** *Programs (17), (19) and (26) are locally equivalent in the following sense. Let  $\theta^r$  be a KKT point of (17) where the constraint  $\mathcal{R}(\theta) \leq r^{-1}$  is active. Then  $\theta^r$  is a critical point for (26) for the value  $\beta(r) = r\mathcal{P}(\theta^r)$ , and it is a KKT point for (19) for the value  $\alpha(r) = \mathcal{P}(\theta^r)/p_1 - 1$ . If  $\theta_\beta$  is a critical point of (26) for parameter  $\beta$ , then it is also a KKT point for (17) with  $r(\beta) =$*

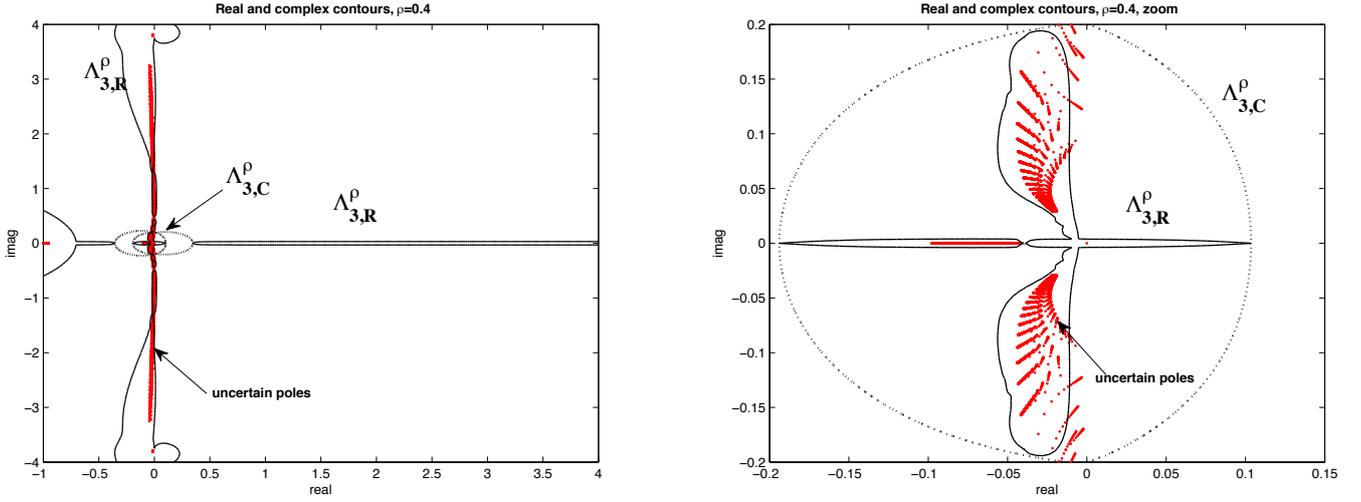


FIGURE 5: Left image shows contours  $\Lambda_{3,\mathbb{R}}$  (continuous) and  $\Lambda_{3,\mathbb{C}}$  (dashed-dotted) at  $\rho = 0.4$  (with  $\rho = \log_{10} q$  for  $q$  as in (24)). The real contour shows the typical beak along the real axis. Right image shows zoom on neighborhood of origin.

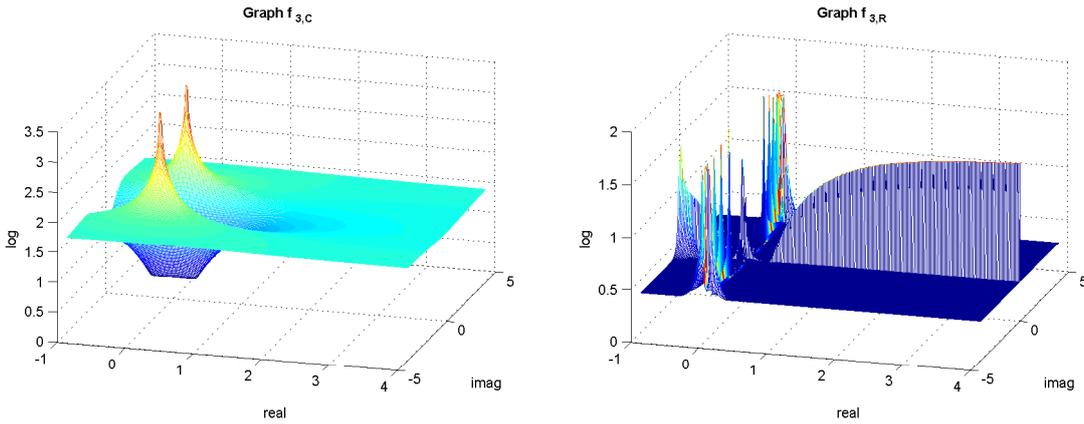


FIGURE 6: Graphs of  $f_{3,\mathbb{C}}$  on the left,  $f_{3,\mathbb{R}}$  on the right. The real contour is again obtained from a very spiky function.

$\mathcal{R}(\theta_\beta)$ , and a KKT point for (19) at  $\alpha(\beta) = \mathcal{P}(\theta_\beta)/p_1 - 1$ . Finally, if  $\theta^{\sharp\alpha}$  is a KKT point of (19) for that value of  $\alpha$ , then it is critical for (26) at  $\beta(\alpha) = (1 + \alpha)p_1/\mathcal{R}(\theta^{\sharp\alpha})$ , and it is a KKT point for (17) with  $r(\alpha)^{-1} = \mathcal{R}(\theta^{\sharp\alpha})$ .

**Proof:** We compare necessary optimality conditions of the three programs. □

In other words, at least locally we have a one-to-one correspondence  $\theta_r \leftrightarrow \theta^\beta \leftrightarrow \theta^{\sharp\alpha}$  as soon as  $\alpha \leftrightarrow r \leftrightarrow \beta$  are in correspondence as above. This gives us the option to solve (19) via (26) and use a line-search in  $\beta$  to steer  $\alpha(\beta)$  toward the desired value  $\alpha$ , respectively  $\mathcal{R}$  to the desired value  $(1 + \alpha)p_1$ . This works often quite satisfactory if objective and constraint are antagonistic. In order to solve the equation  $\alpha(\beta) = \alpha$ , one can use quadratic fitting of several older results  $\alpha(\beta_i) = \alpha_i$  in order to predict a new guess  $\beta$ .

Program (5) respectively (26) can be solved by the recent structured  $H_\infty$ -synthesis function

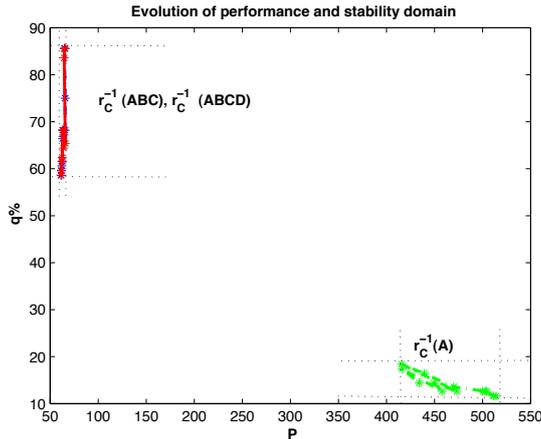


FIGURE 7: Choices  $\mathcal{R} = r_{\mathbb{C}}^{-1}(A)$ ,  $\mathcal{R} = r_{\mathbb{C}}^{-1}(A, B, C)$  and  $\mathcal{R} = r_{\mathbb{C}}^{-1}(A, B, C, D)$  in (7) are compared for the spacecraft example. Every point corresponds to a run of (19) with different  $\alpha$  and  $\mathcal{R}$ .  $r_{\mathbb{C}}^{-1}(A)$  performs badly and gives the cloud on the lower right, with  $\mathcal{P} \geq 400$  and  $q \leq 20$ . The semi-structured stability radii  $r_{\mathbb{C}}(A, B, C)^{-1}$  and  $r_{\mathbb{C}}(A, B, C, D)^{-1}$  produce the cloud on the upper left with good robustness  $q \geq 60$  and a rather mild loss of performance  $\mathcal{P} \geq p_1 = 50$ .

HINFSTRUCT made available through the MATLAB R2010b Prerelease, *Robust Control Toolbox Version 3.5* developed by The MathWorks, Inc., and based on nonsmooth  $H_{\infty}$ -synthesis according to [3]. Evolutions of this method are presented and discussed in [6, 26, 27].

As programs (17), (26) are nonconvex, we use local optimization methods, which in some rare cases may cause discontinuity or rupture of the correspondences  $\theta_r \leftrightarrow \theta^{\beta} \leftrightarrow \theta^{\# \alpha}$  despite a seemingly continuous behavior  $\beta \mapsto r(\beta)$  or  $r \mapsto \beta(r)$ , etc., because we may jump from one branch of local minima to another. This cannot be avoided unless global optimization techniques are applied, but those are usually prohibitively expensive.

It is possible to disregard the non-smoothness of the criteria  $\mathcal{P}$  and  $\mathcal{R}$  and to apply smooth methods like the Matlab function FMINCON. In our present study this did not lead to results of satisfactory quality. Yet another alternative is to use derivative free methods like FMINSEARCH, but these methods, while not asking for derivative information, make the tacit assumption that the functions  $\mathcal{P}, \mathcal{R}$  are smooth. Since this is not the case, one can get surprising results. This problem was already encountered and analyzed in [4]. Search methods for nonsmooth criteria are discussed in [4, 8]. They are usually limited to small problems.

Using a succession of instances of (26) to solve (19) has the advantage that the Matlab function HINFSTRUCT can be used. The proceedings can be organized as follows.

### Algorithm for program (19)

---

Input:  $\alpha > 0$ , plant  $\tilde{P}$ , structure  $K(\theta)$ . Output: Solution  $\theta^{\sharp\alpha}$  of (19),  $\beta = \beta(\alpha)$ .

---

- 1: **Initialize.** Find initial guess  $\beta_1$  with  $\alpha(\beta_1) < \alpha$  and  $\beta_2$  with  $\alpha(\beta_2) > \alpha$ .
  - 2: **Interpolate.** Given two or three consecutive values  $\beta_i$  with corresponding  $\theta^{\beta_i}$ , compute  $\alpha_i = \mathcal{P}(\theta^{\beta_i})/p_1 - 1$ . Use linear or quadratic fit of  $(\alpha_i, \beta_i)$  to obtain new estimate  $\beta$  of  $\beta(\alpha)$ .
  - 3: **Structured  $H_\infty$  synthesis.** Solve program (26) for that value  $\beta$  and obtain  $\theta_\beta$ .
  - 4: **Update.** Compute  $\alpha(\beta) = \mathcal{P}(\theta^\beta)/p_1 - 1$ . If  $\alpha(\beta)$  is sufficiently close to  $\alpha$  stop and return  $\theta^\beta$  as  $\theta^{\sharp\alpha}$  and  $\beta$  as  $\beta(\alpha)$ . Otherwise continue and integrate  $\beta$  among the three best parameter values, dropping the worst one. Loop back to step 2.
- 

In order to solve (26) for fixed  $\beta > 0$  as an instance of (5), we have to form a single channel  $(w, \tilde{w}) \rightarrow (z, \beta\tilde{z})$  in the plant  $\tilde{P}$  such that  $\|T_{(w, \tilde{w}) \rightarrow (z, \beta\tilde{z})}(K)\|_\infty = \max\{\|T_{w \rightarrow z}(K)\|_\infty, \beta\|T_{\tilde{w} \rightarrow \tilde{z}}(K)\|_\infty\}$ .

Using a progress function to solve (17) leads to so-called phase I/phase II methods in the sense of [28]. We have used this approach successfully to address the mixed  $H_2/H_\infty$  problem [7], and the  $H_\infty/H_\infty$  problem [17].

## 7 Posterior analysis with $\mu$ and $\tilde{\mu}$

For small to medium size examples it may be realistic to end the procedure in Algorithm I with a robustness analysis based on the test  $\mu_\Delta(\mathcal{F}_\ell(P, K(\theta))) < q$ , where  $K(\theta)$  is the structured  $H_\infty/H_\infty$  controller with a suitable choice of  $\alpha$  respectively  $r, \beta$ . For a speedy analysis, however, it may be preferable to use the Matlab function `MU`, available in the Robust Control Toolbox. It appears that this function is based on the approximation  $\tilde{\mu}$  of  $\mu$  discussed in [30]. For  $M = \mathcal{F}_\ell(P, K(\theta))$  it can be represented as

$$(27) \quad \tilde{\mu}(M) = \sup_{\omega \in [0, \infty]} \inf\{\mu \geq 0 : M^*(j\omega)S_\omega M(j\omega) + j[T_\omega M(j\omega) - M^*(j\omega)T_\omega] \preceq \mu^2 S_\omega, \\ S_\omega \Delta = \Delta S_\omega, T_\omega \Delta = \Delta T_\omega \forall \Delta \in \mathbf{\Delta}, S_\omega \succeq 0, T_\omega \succeq 0\}.$$

As an alternative, we have implemented the following procedure, which gives a fairly accurate evaluation of  $\tilde{\mu}$ .

### Computation of $\tilde{\mu}$

---

Input:  $M(s)$ ,  $\mathbf{\Delta}$ . Output:  $\tilde{\mu}(M)$ .

---

- 1: **Select frequencies.** Choose a finite subset  $\Omega \subset [0, \infty]$ , possibly including random elements. Initialize  $\mu_0 \geq 0$ . Order  $\Omega$  as  $\omega_1, \dots, \omega_N$ .
- 2: **Main loop.** For  $k = 1, \dots, N$ , given  $\mu_{k-1}$ , check whether the LMI

$$(28) \quad M(j\omega_k)^* S M(j\omega_k) + j[T M(j\omega_k) - M^*(j\omega_k) T] \preceq \mu_{k-1}^2 S, S \succeq 0, T \succeq 0$$

has a solution  $(S, T)$  where  $S, T$  commute with the  $\Delta \in \mathbf{\Delta}$ . If this is the case, put  $\mu_k = \mu_{k-1}$  and loop on with step 2. Otherwise go to step 3.

- 3: **Increase  $\mu$ .** Find smallest  $\mu_k > \mu_{k-1}$  such that (28) has a solution. Then increase counter  $k$  and go back to step 2.
  - 4: Return  $\mu_N$ .
-

$I$ (kg/m <sup>2</sup> )	$\omega_{c_i}$ (rd/s)	$\lambda_i^2$	$\zeta_c$	$\omega_i$ (rd/s)
11200	0.5	6.5	0.003	1.33
	3.3	1.3		3.8
	9.1	1.1		9.5

TABLE 1: Nominal parameter values.

This procedure can be re-started with different random frequencies included in  $\Omega$ , and also the order to run through  $\Omega$  can be changed and adapted. Denoting the optimal  $\mu$  at frequency  $\omega$  by  $\mu_\omega$ , we may obviously speed up things and increase the reliability if we have prior information as to where  $\mu_\omega$  is largest, because then the correction step 3 is required less often and the LMI (28) is solved more easily. On the other hand, as soon as step 3 occurs, it is important to increase  $\mu$  very carefully in order to avoid overestimation of  $\tilde{\mu}$ .

## 8 Case study

We consider the robust control of the roll axis of a geostationary satellite. Model uncertainty is mainly caused by the solar panels, which lead to a large number of flexible modes.

### 8.1 System presentation

A simplified model of the satellite adapted from [16, 19] takes into account the rigid mode of the satellite roll axis, and the first three flexible modes:

$$\begin{aligned} I\ddot{\phi} + p^\top \ddot{\eta} &= u \\ \ddot{\eta}_i + 2\zeta_i\omega_i\dot{\eta}_i + \omega_i^2\eta_i &= -u, \quad i = 1, 2, 3 \end{aligned}$$

Here  $\phi$  is the inertial angular position of the satellite (in rd),  $\eta = (\eta_1, \eta_2, \eta_3)^\top$  is the free mode state vector (in kg · m<sup>2</sup>),  $u$  is the control torque (in Nm) applied to the satellite,  $I$  is the satellite inertia (in kg · m<sup>2</sup>),  $\omega_i$  are the free pulsations (in rd/s),  $\zeta_i = \lambda_i\zeta_c$  are the free damping coefficients,  $p = (\lambda_1^2 - 1, \lambda_2^2 - 1, \lambda_3^2 - 1)^\top$  is the modal participation vector,  $\lambda_i = \omega_i/\omega_{c_i}$  are the free cantilever pulsation ratios,  $\omega_{c_i}$  are the cantilever pulsations (in rd/s), and  $\zeta_c$  is the common cantilever damping. The system is represented schematically in figure 8, the nominal parameter values are gathered in table 1.

The maximal uncertain parameter vector is  $\delta = (\omega_{c_1}, \omega_{c_2}, \omega_{c_3}, \lambda_1^2, \lambda_2^2, \lambda_3^2, I) \in \mathbb{R}^7$ , but the authors of [19] show that one can concentrate on three uncertain parameters  $\omega_{c_1}, \lambda_1^2, I$ , so the vector of uncertain parameters in our testing is  $(\delta_1, \delta_2, \delta_3) = ((\omega_{c_1} - \bar{\omega}_{c_1})/\bar{\omega}_{c_1}, (\lambda_1^2 - \bar{\lambda}_1^2)/\bar{\lambda}_1^2, (I - \bar{I})/\bar{I})$ , where over-lined entities are nominal values. It turns out that two of the three uncertainties are repeated, so that the matrix  $\Delta \in \mathbf{\Delta}$  has the structure  $\Delta = \text{diag}(\delta_1, \delta_1, \delta_2, \delta_2, \delta_3) \in \mathbb{R}^{5 \times 5}$ . Multipliers  $S \in \mathbb{R}^{5 \times 5}$  which commute with  $\Delta \in \mathbf{\Delta}$  have therefore two  $2 \times 2$  blocks and a  $1 \times 1$  block. The scheme of the uncertain model is given in figure 9. This includes the block  $[q_1, \dots, q_5]^\top = \Delta[p_1, \dots, p_5]^\top$  in (10).

The goal is to stabilize the system robustly over a set (9) of parameter variations, and to regulate the roll angle (in rd) of the space vehicle to obtain a short settling time, reasonable overshoot, and good torque disturbance rejection.

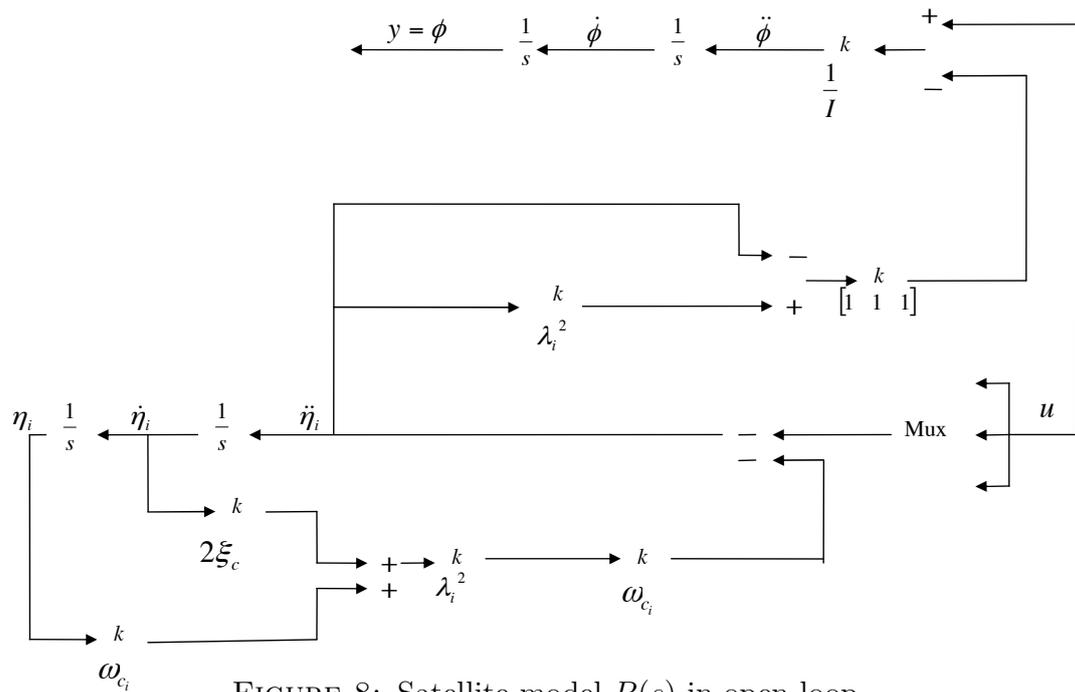


FIGURE 8: Satellite model  $P(s)$  in open loop.

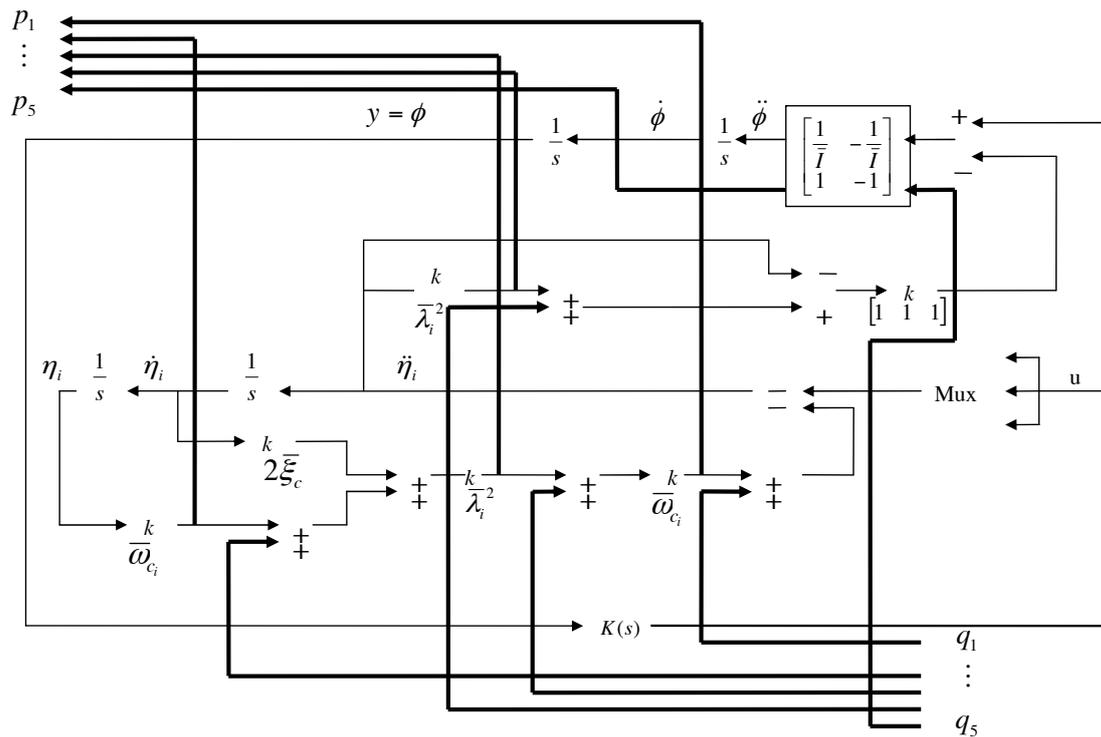


FIGURE 9: Uncertain model  $M(s)$  for satellite.

## 8.2 Nominal design

For synthesis we have used the configuration of figure 10, with two choices of the low pass filter  $F$  and the coefficient  $\epsilon$ :

$$(29) \quad \text{case 1: } F_1(s) = \frac{s+0.1}{(s+0.01)^2} \text{ and } \epsilon_1 = 1.0 \quad \text{case 2: } F_2(s) = \frac{1}{s+0.001} \text{ and } \epsilon_2 = 0.001.$$

The controlled input  $w$  is a reference  $\phi_{\text{ref}}$  on the roll angle, the regulated output is  $z = (z_1, z_2)$  with the weighing described above. In case 1 the system has 10 states (8 system + 2 filter), in the second case we have 9 states (8 system + 1 filter).

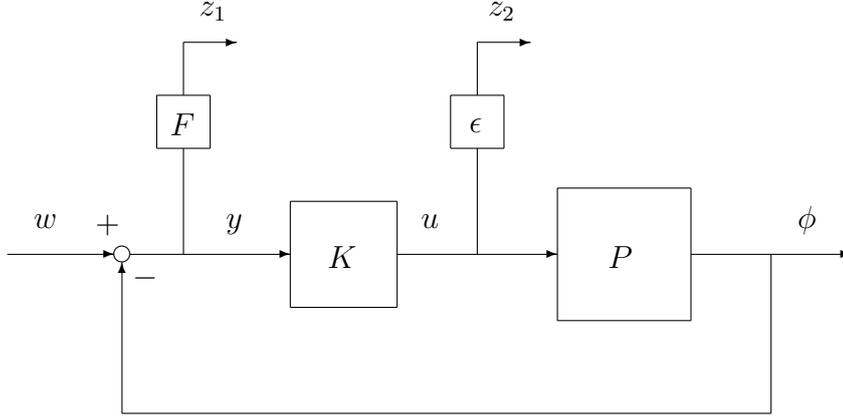


FIGURE 10: The feedback scheme.

A second aspect of this study is that the satellite has to be regulated with a low order controller, because full order controllers with orders  $n_K = 9$  respectively  $n_K = 10$  are too clumsy to be of practical use. In consequence we have synthesized controllers of reduced order  $n_K = 5$ . As the system is SISO, this leads to  $n = 36$  unknown variables in programs (5), (19) or (7). Synthesis within the structure of 5th order controllers leads to nominal performance  $p_1^* = 61.48$  in case 1, and  $p_2^* = 2.94$  in case 2.

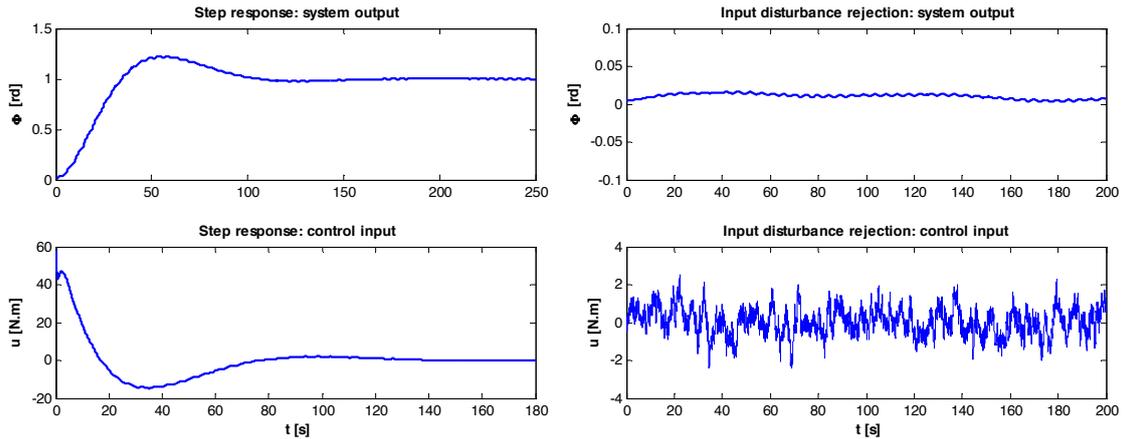


FIGURE 11: Time domain responses of slow 5th-order controller  $K_1$ .

The 5th-order  $H_\infty/H_\infty$  controllers for the two scenarios in (29), computed via program (26), will be denoted  $K_1$  and  $K_2$ . As can be seen in Figures 11, 12 the step response for  $K_2$  is considerably faster than the one for  $K_1$ , so that the frequency band pass of the closed-loop system is increased,

which accounts for the fact that disturbances are badly rejected. Figures 11,12 also feature the control inputs generated by  $K_1, K_2$ .

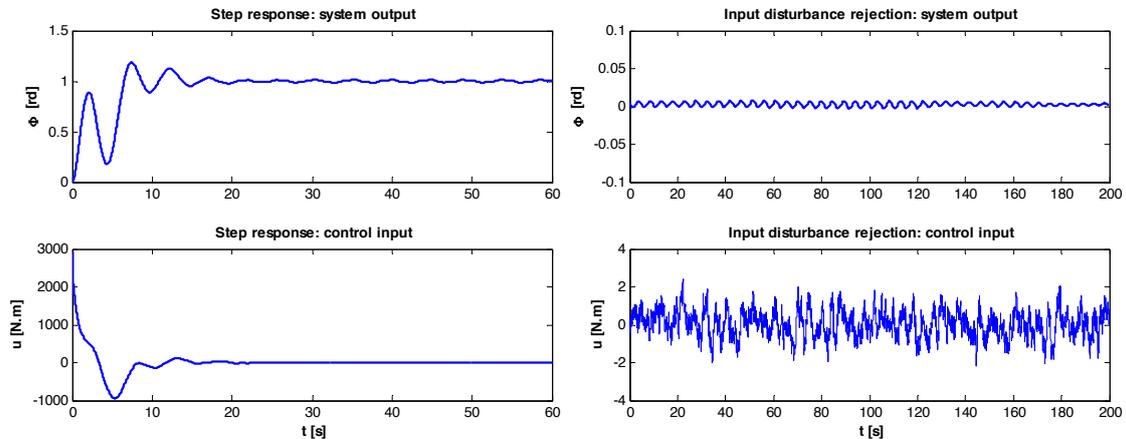


FIGURE 12: Time domain responses of fast 5th-order controller  $K_2$ .

In order to assess parametric robustness, we measure the stability cube  $Q$  by representing it as  $Q = qQ_0$ , where  $\max\{\sigma_1(\Delta) : \Delta \in Q_0\} = 1$ . This allows to display  $q$  as a curve against parameter  $\alpha$  in (19), allowing comparison with  $\mathcal{R} = r_{\mathcal{C}}^{-1}$  and  $\mathcal{P}$ . We refer to  $q$  as the percentage of parametric robust stability, where  $q = 100\%$  corresponds to stability over  $Q_0$ . For  $K_2$  we initially achieve only  $q_2 = 0.15\%$ , while the nominal  $K_1$  gives  $q_1 = 58.6\%$ . Our experiment shows that program (7), respectively (26), allows to increase parametric robustness  $q$  for both scenarios, despite their completely different nominal behavior.

### 8.3 Enhanced parametric robustness

In Figure 13 the results for scenario 1 are shown. As parameter  $\alpha$  in (19) varies from 0 to 250%, we can see a global increase of  $\mathcal{P}$ , which means a gradual loss of performance. At the same time robustness  $\mathcal{R} = r_{\mathcal{C}}^{-1}$  decreases as we increase  $\alpha$  from 0 up to  $\alpha = 100\%$ , which corresponds to an increase of the semi-structured distance to instability. For values  $\alpha$  larger than 100% robustness  $\mathcal{R}$  stagnates without significant increase. Parametric robustness  $q$  increases initially, to stagnate from  $\alpha = 25\%$  onwards. As we zoom on the range  $\alpha \in [0\%, 25\%]$  (upper right) and again on  $\alpha \in [0\%, 7\%]$  (lower image), we see that program (26) is useful up to  $\alpha \approx 7\%$ , where it leads to a steady increase in parametric robustness, with a strong increase around  $\alpha = 5\%$ .

For scenario 2 the same procedure was followed, the results being shown in Figure 15. Increase of  $\alpha$  allows to increase parametric robustness from  $q = 0.15\%$  up to  $q = 16.4\%$ , with a snapshot at  $q = 11.6\%$  shown in between. Performance degrades moderately as  $\alpha$  increases, while robustness  $\mathcal{R} = r_{\mathcal{C}}^{-1}$  has a dramatic jump at the beginning, to decrease moderately from  $\alpha = 3\%$  onwards. The size of the stability domain  $qQ_0$  is shown in Figure 15 for the 3 snapshots.

Finally, using the method of section 7, we use the obtained controller  $K_5$  to compare  $\mu, \tilde{\mu}$  and its Matlab approximation MU. The results are shown in Figure 16. In this study  $\Omega$  contained 2000 frequencies in the range  $[10^{-0.35}, 10^{1.08}]$ . The LMI in step 2 of the algorithm of section 7 was solved with the Matlab function MINCX. We initialized  $\mu_0 = 1$  and increment  $\mu_k$  by 0.01 in step 3 until a new  $\mu_{k+1}$  satisfying (28) was found. For the result in Figure 16 our method needed 655s CPU

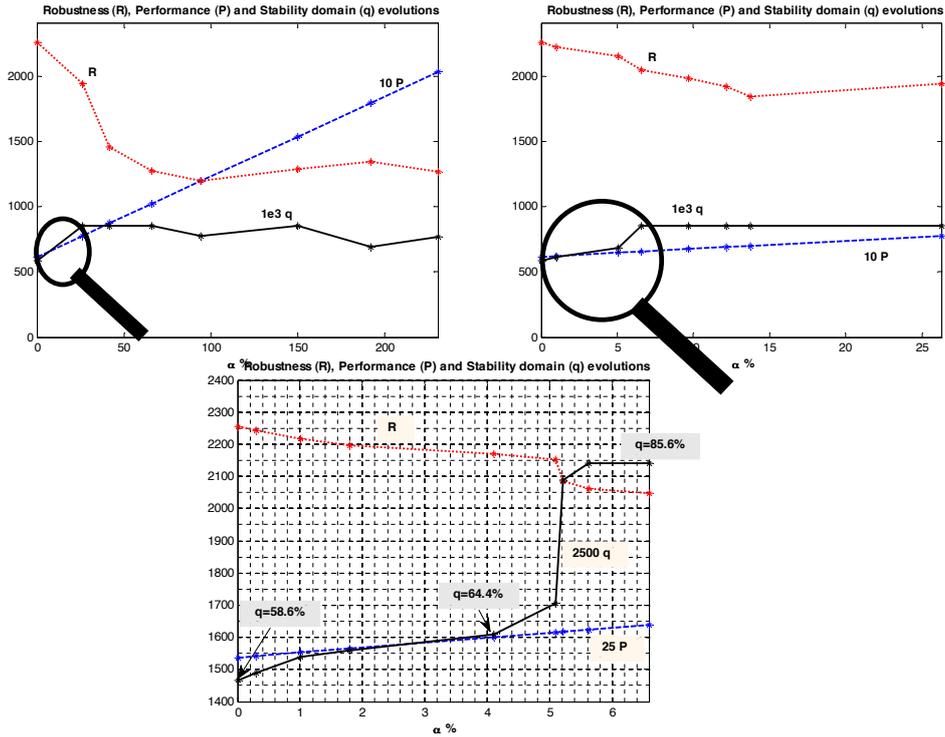


FIGURE 13: Program (19) solved for various values  $\alpha$ . Performance  $\mathcal{P}$ , robustness  $\mathcal{R}$ , and size of the stability cube  $q$  are displayed against  $\alpha$  for scenario 1 in (29). Performance degrades as  $\alpha$  increases, robustness improves, and  $q$  increases with a strong slope around  $\alpha = 5.2\%$ .

to compute  $\tilde{\mu}$ , while MU required 112s CPU. We have observed that MU is indeed made rather for speed than precision, so the discrepancy seen in Figure 16 is rather typical. It is also interesting to notice that contrary to what is often claimed,  $\tilde{\mu}$  almost always differs from  $\mu$ , which means that even with multipliers  $S_\omega, T_\omega$  some conservatism cannot be avoided. This is important because it can be understood as a disclaimer to use  $\tilde{\mu}$  directly in optimization. Using it as a posterior evaluation as discussed in section 7 seems preferable.

## 9 Conclusion

Multi objective  $H_\infty/H_\infty$  synthesis was used to enhance robustness in closed-loop in the presence of real uncertain parameters in the system. The synthesized control laws can in addition be structured, which includes low-order controllers, decentralized and PID or observer-based controllers, control architectures including set-point or washout filters, feed forward, and much else. We have tested an approach which minimizes the nominal  $H_\infty$  performance objective subject to a constraint  $r_{\mathcal{C}} \geq r$  guaranteeing a non-negligible distance to instability of the closed-loop system. Contour plots were used to investigate the degree of conservatism in our approach. The method was tested numerically to control the roll axis of a flexible satellite with 3 uncertain parameters.

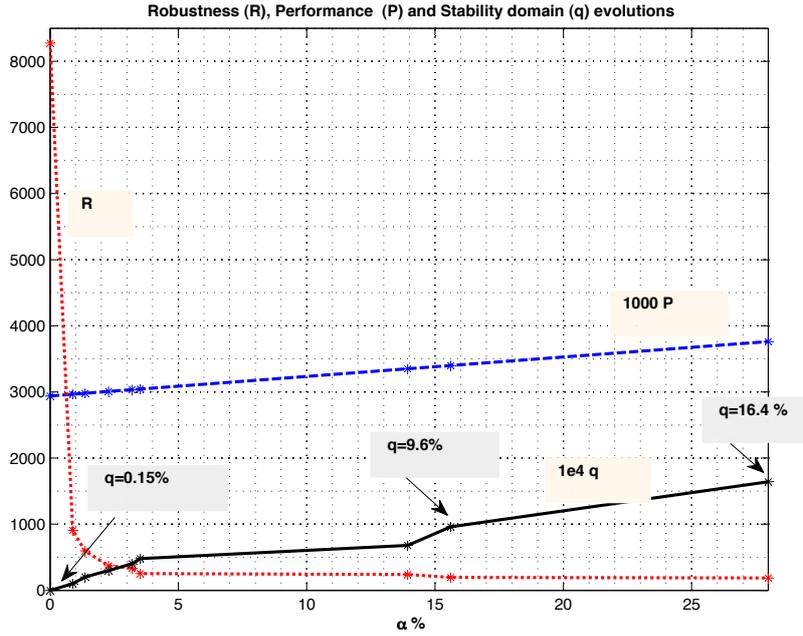


FIGURE 14: Performance  $\mathcal{P}$ , robustness  $\mathcal{R}$ , and size of the stability cube  $q$  for controller  $K_2$  displayed against  $\alpha$ . Choices of  $\alpha$  to increase  $q$  from  $q = 0.15\%$  on the left to  $q = 16.4\%$  on the right can be read off.

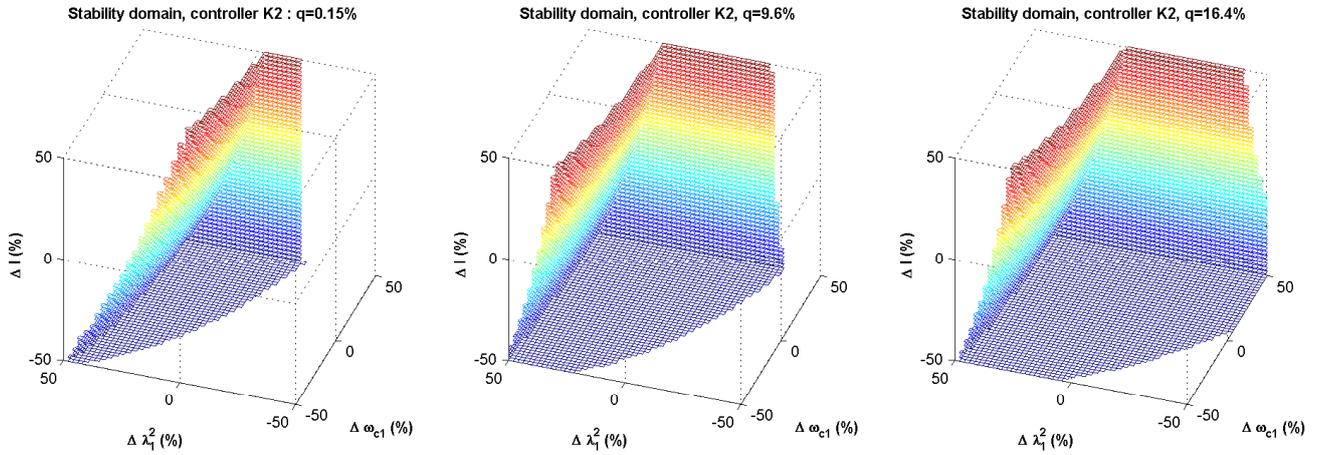


FIGURE 15: Stability cube for  $K_2$  is shown for  $q = 0.15\%$  upper left,  $q = 9.6\%$  upper right,  $q = 16.4\%$  lower left. This corresponds to the situation in figure 14.

## Acknowledgement

The authors are grateful for financial support from Agence Nationale de Recherches (ANR) under contract *Guidage*, from Fondation de Recherches pour l'Aéronautique et l'Espace (FNRAE) under contract *Survool*, and from Fondation d'Entreprise EADS (F-EADS) under contract *Technicom*.

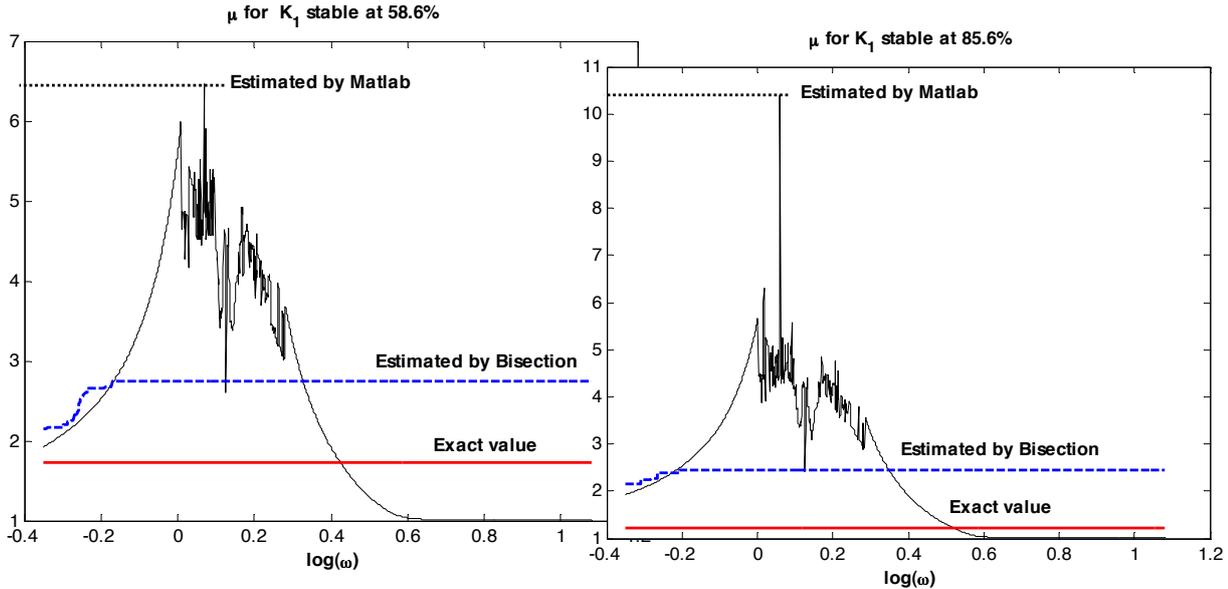


FIGURE 16: Robustness analysis of controller  $K_5$  for two values of  $q$ . The exact value  $\mu_\Delta$ , which corresponds to  $r_\Delta^{-1}$  is the lowest curve. Our estimation  $\tilde{\mu}$ , which corresponds to  $\tilde{r}_C^{-1}$  is the dashed curve, and the result of the Matlab function MU is dotted.

## References

- [1] P. APKARIAN. *Nonsmooth  $\mu$ -synthesis*. Int. J. Rob. Nonlin. Control, to appear.
- [2] P. APKARIAN, L. HOSSEINI-RAVANBOD, D. NOLL. *Time domain constrained structured  $H_\infty$  synthesis*. International Journal of Robust and Nonlinear Control, vol. 21, no. 2, 2011, pp. 197 – 217.
- [3] P. APKARIAN, D. NOLL. *Nonsmooth  $H_\infty$  synthesis*. IEEE Transactions on Automatic Control, vol. 51, no. 1, 2006, pp. 71 – 86.
- [4] P. APKARIAN, D. NOLL. *Controller design via nonsmooth multi-directional search*. SIAM J. Control and Optimization, vol. 44, no. 6, 2006, pp. 1923 – 1949.
- [5] P. APKARIAN, D. NOLL. *Nonsmooth optimization for multidisk  $H_\infty$  synthesis*. European Journal of Control, vol. 12, no. 3, 2006, pp. 229 – 244.
- [6] P. APKARIAN, D. NOLL, O. PROT. *A trust region spectral bundle method for nonconvex eigenvalue optimization*. SIAM Journal on Optimization, vol. 19, no. 1, 2008, pp. 281 – 306.
- [7] P. APKARIAN, D. NOLL, A. RONDEPIERRE. *Mixed  $H_2/H_\infty$  control via nonsmooth optimization*. SIAM Journal on Control and Optimization, vol. 47, no. 3, 2008, pp. 1516 – 1546.
- [8] C. AUDET, J.E. DENNIS JR. *Mesh adaptive direct search algorithms for constrained optimization*. SIAM Journal on Optimization, vol. 17, no. 1, 2006, pp. 188-217.
- [9] B. BERNHARDSSON, A. RANTZER, L. QIU. *On real perturbation values and real quadratic forms in a complex vector space*. Linear Algebra and its Applications, vol. 270, no. 1 - 3, 1998, pp. 131 – 154.

- [10] V. D. BLONDEL, J.N. TSITSIKLIS. *NP-hardness of some linear control design problems*. SIAM J. of Control and Opt., 35:6, pp. 2118-2127, 1997.
- [11] V. D. BLONDEL, J. N. TSITSIKLIS. *A survey of computational complexity results in systems and control*. Automatica, vol. 36, no. 9, 2000, pp. 1249–1274.
- [12] V. BOMPART, D. NOLL, P. APKARIAN. *Second-order nonsmooth optimization for  $H_\infty$ -synthesis*. Numerische Mathematik, vol. 107, 2007, pp. 433 – 454.
- [13] S. BOYD, C. BARRATT. *Linear controller design. Limits of performance*. Prentice Hall, 1991.
- [14] J. BURKE, A. LEWIS, M. OVERTON. *Optimization and pseudospectra, with applications to robust stability*. SIAM Journal on Matrix Analysis and Applications, vol. 25, no1, 2003, pp. 80–104.
- [15] M.K.H. FAN, A.L. TITS, J.C. DOYLE. *Robustness in the presence of mixed parametric uncertainty and unmodelled dynamics*. IEEE Trans. Autom. Control, vol. AC-36, no. 1, pp. 25 – 38.
- [16] B. FRAPARD, C. CHAMPETIER.  *$H_\infty$  techniques: from research to industrial applications*. Proceedings of the 3rd International Conference on Spacecraft Guidance, Navigation and Control Systems, ESTEC, Noordijk, The Netherlands, 26-29 November 1996.
- [17] M. GABARROU, D. ALAZARD, D. NOLL. *Structured flight control law design using non-smooth optimization*. 18th IFAC Symposium on Automatic Control in Aerospace (ACA 2010). 6 - 10 septembre 2010, Nara, Japon.
- [18] P. GAHINET, P. APKARIAN. *A linear matrix inequality approach to  $H_\infty$  control*. Int. Journal of Robust and Nonlinear Control, pp. 421-448, 1994 .
- [19] M. GAUVRIT, D. ALAZARD. *Parametric worst-case analysis by PRABI method: application to flexible space structures*. 2nd IFAC symposium on Robust Control Design, Budapest, 1997.
- [20] D. HINRICHSSEN, B. KELB. *Spectral value sets: a graphical tool for robustness analysis*. Systems and Control Letters, vol. 21, 1993, pp. 127 – 136.
- [21] D. HINRICHSSEN, A.J. PRITCHARD. *Stability radii of linear systems*. Systems and Control Letters, vol. 7 (1986) 1 - 10.
- [22] D. HINRICHSSEN, A.J. PRITCHARD. *Stability radius for structured perturbations and the algebraic Riccati equation*. Systems and Control Letters, vol. 8 (1986) 105 - 113.
- [23] D. HINRICHSSEN, A.J. PRITCHARD. *Mathematical System Theory I. Modelling, State Space Analysis, Stability and Robustness*. Springer Verlag 2005.
- [24] M. KAROW, E. KOKIOPOULOU, D. KRESSNER. *On the computation of structured singular values and pseudospectra*. Systems Control Lett., 59(2):122-129, 2010.
- [25] O. MERINO, H. DYM, J.W. HELTON. *Algorithms for solving multidisk problems in  $H_\infty$  optimization*. Proceedings of the IEEE Conference on Decision and Control, 1999.

- [26] D. NOLL. *Cutting plane oracles to minimize nonsmooth and nonconvex functions*. Set-Valued and Variational Analysis, vol. 18, no. 3-4, 2010, pp. 531 – 568.
- [27] D. NOLL, O. PROT, A. RONDEPIERRE. *A proximity control algorithm to minimize nonsmooth and nonconvex functions*. Pacific Journal of Optimization, vol. 4, no. 3, 2008, pp. 569-602.
- [28] E. POLAK. *Optimization: Algorithms and Consistent Approximations*. Springer Verlag, New York, 1997.
- [29] L. QIU, B. BERNHARDSSON, A. RANTZER, E.J. DAVIDSON, P.M. YOUNG, J.C. DOYLE. *A formula for computation of the real stability radius*. Automatica, vol. 31, no. 6 (1995), 879 - 890.
- [30] J. SREEDHAR, P. VAN DOOREN, A.L. TITS. *A fast algorithm to compute the real structured stability radius*. Int. Series of Numer. Math., Birkhäuser, pp. 219 – 230, 1996.
- [31] L.N. TREFETHEN. *Computation of pseudospectra*. Acta Numerica, vol. 8, 1999.
- [32] L.N. TREFETHEN, M. EMBREE. *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton University Press 2005.
- [33] K. ZHOU. *Essentials of Robust Control*. Prentice Hall, 1998.