HAL
open science

The H∞ Control Problem is Solved

P. Apkarian, Dominikus Noll

▶ To cite this version:

P. Apkarian, Dominikus Noll. The H∞ Control Problem is Solved. Aerospace Lab, 2017, 13, p. 1-11. 10.12762/2017.AL13-01 . hal-01653161

HAL Id: hal-01653161
https://hal.science/hal-01653161

Submitted on 1 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

P. Apkarian
(ONERA)

D. Noll
(Institut de Mathématiques de Toulouse, Université Paul Sabatier)

E-mail: pierre.apkarian@onera.fr

DOI: 10.12762/2017.AL13-01

The H_∞ Control Problem is Solved

The H_∞ control problem was posed by G. Zames in 1981 [1], and various attempts to address it had been made over the years. Ultimately, in 2006, the authors presented their solution, which is based on a tailored non-smooth optimization technique [2]. In this treatise we present the rationale of H_∞ control, give a brief history, and recall the milestones reached before our 2006 solution. We clarify why our novel approach is welcomed in the high-tech and aerospace industry. Recent MATLAB functions, `hinfstruct` and `systune`, based on work by Apkarian, Noll and Gahinet (The MathWorks) are available in the Robust Control Toolbox, since R2010b and R2012b respectively.

What has Rosetta got to do with H_∞-control?

The *Rosetta space probe* developed by the European Space Agency was launched on March 2, 2004 with an Ariane-5 launcher, and its lander module *Philae* was successfully placed on the comet 67P/Churyumov-Gerasimenko by November 12, 2014. What is less known is that in 2011 an unexpected off-pointing of the satellite was detected. In-depth on-ground analysis revealed a loss of efficiency in one of the thrusters. It then became evident that the successful accomplishment of the mission hinged on re-designing the controllers. New sophisticated attitude controllers were synthesized by Airbus Defense and Space in March 2014, using the multi-model features of a novel controller synthesis technique called `hinfstruct`. These new controllers were uploaded in May 2014, just before engaging the final maneuver to get closer to the comet 67P/Churyumov-Gerasimenko [35]. The novel control design tool that was behind this had been pioneered by the authors [2] between 2004 and 2006. It became available to control engineers between 2006 and 2010 via the MATLAB functions `hinfstruct` and `systune`. The mathematical principle underlying these tools is the H_∞-rationale, which we will explain in this treatise.

We mention that a change of paradigm in control engineering is currently underway, where our novel structured H_∞-control design technique is being adopted by leading aerospace industries. For instance, Dassault and ONERA [36] use the H_∞-technique in tandem with the MORE software [37] to test new strategies for anti-vibration control of civil aircraft. Design of new atmospheric flight pilots for the Ariane launcher is being investigated by Airbus Safran Launchers, CNES and ONERA. Other applications include control of flexible satellites by Thales Alenia Space and ADS, inertial line of sight stabilization by SAGEM [34], the design of structured estimators for microsatellites by CNES [33], motor torque control in haptics by the CEA LIST robotics [32], and the list could be continued.

A mathematical principle, the H_∞-paradigm, has found its way into control engineering practice. We investigate its rationale, and gauge the potential of the new method for the high-tech industry.

The H_∞ control problem

The H_∞-problem was framed by G. Zames in two plenary talks at the IEEE CDC in 1976 and the Allerton Conference in 1979, and was posed formally in his 1981 paper [1]. However, the origins of the H_∞-problem are much older and date back to the 1960s, when Zames discovered the small gain theorem [4]. After more than 30 years, the H_∞-problem was "solved" by P. Apkarian and D. Noll in 2006 [2] in a sense that is defined hereafter. Note a related, though very different, technique of stochastic nature has been developed in [3].

In this section we introduce the H_∞-control problem formally, discuss its rationale, and present the context leading to our 2006 solution.

Some history

In their seminal 1989 paper [5], Doyle, Glover, Khargonekar and Francis show that the H_∞ problem requires the solution of two algebraic Riccati equations (AREs). Doyle [6] discusses how this

milestone is reached and mentions an earlier 1984 solution. In 1994, P. Gahinet and P. Apkarian give a solution [7] of the H_∞ problem by reducing it to a linear matrix inequality (LMI), the 1995 solution. How can a problem be solved several times? What do we mean when we say that we solved the problem in 2006 [2], when the 1984, 1989, and 1995 solutions existed already?

Formal statement of the problem

The H_∞ control problem can be stated as follows. Given a real rational transfer matrix $P(s)$, called the *plant*, and a space $\mathcal{K}$ of real rational transfer matrices $K(s)$, called the *controller space*, characterize and compute an optimal solution $K^* \in \mathcal{K}$ to the following optimization program

$$\begin{array}{ll} \text{minimize} & \| T_{w\to z}(P,K) \|_\infty \\ \text{subject to} & K \text{ stabilizes } P \text{ internally} \\ & K \in \mathcal{K} \end{array} \qquad (1)$$

Here, the objective function is the H_∞-norm of the closed-loop performance channel $T_{w\to z}(P,K)$, see Figure 1. As we shall see, the choice of the controller space $\mathcal{K}$ in (1) is the key to a proper understanding of the problem.

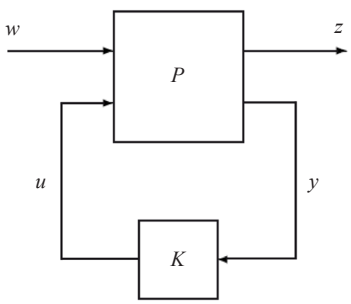

Figure 1 – Standard closed-loop LFT model

Let us recall the notions used to formulate (1). The plant $P(s)$ has a state-space representation of the form

$$P: \begin{cases} \dot{x} = Ax + B_1 w + B_2 u \\ z = C_1 x + D_{11} w + D_{12} u \\ y = C_2 x + D_{21} w + D_{22} u \end{cases} \qquad P(s): \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \qquad (2)$$

where $x \in \mathbb{R}^{n_p}$ is the state, $u \in \mathbb{R}^{n_u}$ the control, $y \in \mathbb{R}^{n_y}$ the measured output, $w \in \mathbb{R}^{n_w}$ the exogenous input, and $z \in \mathbb{R}^{n_z}$ the regulated output. Similarly, $K(s)$ has the state-space representation

$$K: \begin{cases} \dot{x}_K = A_K x_K + B_K y \\ u = C_K x_K + D_K y \end{cases} \qquad K(s): \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right] \qquad (3)$$

where $x_K \in \mathbb{R}^k$ is the state of K. As soon as $D_{22} = 0$, the closed-loop transfer channel $T_{w\to z}(P,K)$ in (1) has the state-space representation

$$T_{w\to z}(P,K): \begin{bmatrix} A(K) & B(K) \\ C(K) & D(K) \end{bmatrix} \qquad (4)$$

where

$$A(K) = \begin{bmatrix} A + B_2 D_K C_2 & B_2 C_K \\ B_K C_2 & A_K \end{bmatrix}, B(K) = \begin{bmatrix} B_1 + B_2 D_K D_{12} \\ B_K D_{21} \end{bmatrix}, C(K) = \text{etc.} \qquad (5)$$

and where the state dimension is now $n_p + k$. Finally, for a stable real rational transfer function $T(s)$, the H_∞-norm in (1) is defined as

$$\|T\|_\infty = \max_{\omega \in \mathbb{R}} \bar{\sigma}(T(j\omega)) \qquad (6)$$

where $\bar{\sigma}(M)$ is the maximum singular value of a complex matrix M.

With these notations, we can now give the first explanation. The 1984, 1989 and 1995 solutions of the H_∞ problem (1) are all obtained within the space $\mathcal{K}_{full}$ of full-order controllers

$$\mathcal{K}_{full} = \{K : K \text{ has form (3) with size}(A_K) = \text{size}(A)\}$$

Observe that in $\mathcal{K}_{full}$ *all* entries in A_K, B_K, C_K, D_K are free variables. Altogether, there are $N := n_p^2 + n_p(n_y + n_u) + n_y n_u$ degrees of freedom and we have

$$\mathcal{K}_{full} \cong \mathbb{R}^N$$

In particular, $\mathcal{K}_{full}$ is the largest controller space that we could use in (1)[1]. Finding a solution within $\mathcal{K}_{full}$ is therefore easier. In particular, with $\mathcal{K}_{full}$ as the controller space, (1) is convex, as shown in [7]. When *smaller* and more practical controller spaces $\mathcal{K}$ are chosen, Problem (1) is much harder to solve. Our 2006 solution addresses these difficult cases.

Solutions of the H_∞-control problem in the 1980s and 1990s refer to the full-order case, which is essentially convex.

The rationale

After closing the loop in the feedback scheme (1) we may consider the closed-loop system as a linear operator $T_{w\to z}(P,K)$ mapping input w to output z. If K stabilizes P internally, that is, if $T_{w\to z}(P,K)$ in (5) is stable, then $T_{w\to z}(P,K)$ maps $w \in L^2$ into $z \in L^2$. The H_∞-norm (6) is then nothing else but the L^2- L^2-operator norm, that is,

$$\|T\|_\infty = \sup_{w \neq 0} \frac{\|Tw\|_2}{\|w\|_2} = \sup_{w \neq 0} \frac{\|z\|_2}{\|w\|_2}$$

In other words, for a closed-loop channel $w \to z$ the norm squared $\gamma^2 = \|T_{w\to z}(P,K)\|_\infty^2$ is the factor by which the energy of the input signal is amplified in the output. Input w with energy $\|w\|_2^2$ will produce output z with energy $\|z\|_2^2$ no greater than $\gamma^2 \cdot \|w\|_2^2$, as long as controller K is used. The optimization program (1) strives to find the controller $K^* \in \mathcal{K}$ for which this amplification factor γ is smallest.

In a closed loop with controller K, the input w with energy $\|w\|_2^2$ creates output z with energy $\|z\|_2^2 \le \gamma^2 \|w\|_2^2$, where $\gamma = \|T_{w\to z}(P,K)\|_\infty$. The same relation holds for power signals $w \to z$, i.e., power is amplified by no more than γ^2.

[1] Using even larger state dimensions does not lead to anything new.

This can obviously be very useful. All that we have to do is to find communication channels $w \to z$, where the smallness of answer z to question w tells us something meaningful about the system.

We now give the typical context of *loopshaping*, where this idea is used. The standard control scheme (see Figure 2) features the open-loop system G, the controller K, the measured output y, the control signal u, and the tracking error e. Red signals are inputs, $n_s =$ sensor noise, $d =$ disturbance or process noise, and $r =$ reference signal for y, sometimes called a command. The blue signals are specifically chosen outputs, $\tilde{e} = W_e e$, $\tilde{u} = W_u u$, $\tilde{y} = W_y y$.

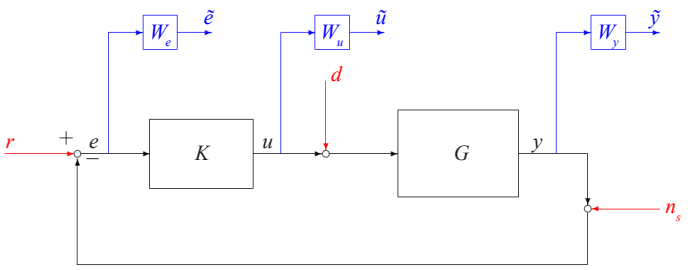

Figure 2 – Standard control scheme

This is a special case of Figure 1, where $w = (r, d, n_s)$ is the input, $z = (\tilde{e}, \tilde{u}, \tilde{y})$ is the output, and where plant G regroups G and the filters W_e, W_u, W_y. The filters may be dynamic, which adds new states to the plant P.

What are useful transfer functions from red to blue? For instance, the transfer from reference r to tracking error e

$$T_{r\to e}(K) = (I + GK)^{-1}$$

is a typical performance channel, because it describes how fast the system follows the reference r. Since one typically wants to track only in the low frequency range, W_e is a low-pass filter. Now, the smallness of the norm

$$\left\| T_{r\to\tilde{e}}(K) \right\|_\infty = \left\| W_e (I + GK)^{-1} \right\|_\infty$$

means that the low frequency component $\tilde{e}$ of the tracking error e becomes small as a result of optimization, so y follows the reference input r in low frequency.

Next consider a typical robustness channel. For instance, the influence of sensor noise n_s on the control signal u. Noise is typically of high frequency, but that should not lead to high frequency components in u, as this bears the risk, for example, of actuator fatigue. Therefore, W_u is typically a high-pass filter and $\tilde{u}$ are high frequency components of u. We find

$$T_{n_s\to\tilde{u}}(K) = -W_u (I + KG)^{-1} K$$

and $\left\| T_{n_s\to\tilde{u}}(K) \right\|_\infty$ puts a cost on high frequency components in u. If program (1) is successful, it will furnish an optimal $K^* \in \mathcal{K}$ that makes this cost as small as possible, thereby building robustness to sensor noise into the system.

To conclude, we can see that, depending on the specific application, there will be several performance and robustness channels. In its basic form, (1) requires fixing a single connection $w \to z$, but in Section 5 we will show how to solve a multi-objective problem with several H_∞-channels.

Setting up the performance channel $w \to z$ in (1) could be interpreted as putting a cost on undesirable behavior of the closed-loop system.

Controller structures

The reason why the H_∞ theory of the 1980s failed to take hold in practice is quickly explained. Controllers computed via algebraic Riccati equations are full order, or *unstructured*. However, for various reasons, practitioners prefer simple controllers like PIDs, or control architectures combining PIDs with filters, and such controllers are *structured*.

The discrepancy between H_∞ theory and control engineering practice is highlighted, for example, by PID control. Until 2010 PID controllers had to be tuned instead of optimized, because software for H_∞-PID control was not available.

During the 1990s and early 2000s a new approach to controller design based on linear matrix inequalities (LMIs) was developed. Unfortunately, LMIs have essentially the same shortcomings as AREs: H_∞ controllers computed via LMIs are still unstructured. The situation only started to improve when, in the late 1990s, the authors pioneered the investigation of feedback controller synthesis via bilinear matrix inequalities (BMIs). While the LMI euphoria was still in full progress, we recognized that what was needed were algorithms that would allow structured controllers to be synthesized. Here is the formal definition of structure (see [2]).

Definition 1
A controller K of the form (3) *is called* structured *if the state-space matrices A_K, B_K, C_K, D_K depend smoothly on a design parameter vector κ varying in some parameter space $\mathbb{R}^n$, or in a constrained subset of $\mathbb{R}^n$.*

In other words, a controller structure $K(\cdot)$, or $K(\kappa)$, consists of four smooth mappings $A_K(\cdot): \mathbb{R}^n \to \mathbb{R}^{k\times k}$, $B_K(\cdot): \mathbb{R}^n \to \mathbb{R}^{k\times n_y}$, $C_K(\cdot): \mathbb{R}^n \to \mathbb{R}^{n_u\times k}$, and $D_K(\cdot): \mathbb{R}^n \to \mathbb{R}^{n_u\times n_y}$.

It is convenient to indicate the presence of structure in K by the notation $K(\kappa)$, where κ denotes the free parameters. In the MATLAB functions `hinfstruct` *or* `systune` *one refers to κ as the vector of tunable parameters.*

Three basic examples with structure

The structure concept is best explained by examples. The transfer function of a realizable PID controller is of the form

$$K(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{1 + T_f s} = d_K + \frac{r_i}{s} + \frac{r_d}{s + \tau}, \qquad (7)$$

where $d_K = k_p + k_d / T_f$, $\tau = 1/T_f$, $r_i = k_i$, $r_d = -k_d T_f^2$. Realizable PIDs may therefore be represented in state-space form

$$K_{pid}(\kappa): \begin{bmatrix} 0 & 0 & r_i \\ 0 & -\tau & r_d \\ 1 & 1 & d_K \end{bmatrix} \tag{8}$$

where $\kappa = (r_i, r_d, d_K, \tau) \in \mathbb{R}^4$ is tunable. As we can see,

$$A_K(\kappa) = \begin{bmatrix} 0 & 0 \\ 0 & -\tau \end{bmatrix}, B_K(\kappa) = \begin{bmatrix} r_i \\ r_d \end{bmatrix}, C_K(\kappa) = [1\,1], D_K(\kappa) = d_K$$

If we use the PID structure (8) within the H_∞ framework (1), we compute an H_∞ PID controller, that is, a PID controller that minimizes the closed-loop H_∞-norm among all internally stabilizing PID controllers:

$$\left\| T_{w\to z}\left(P, K_{pid}^*\right) \right\|_\infty \le \left\| T_{w\to z}\left(P, K_{pid}\right) \right\|_\infty$$

The controller space for this structure is

$$\mathcal{K}_{pid} = \left\{ K_{pid}(\kappa) : \text{as in (8)}, \kappa = (r_i, r_d, d_K, \tau) \in \mathbb{R}^4 \right\}$$

The fact that PID is a structure in the sense of Def. 1 means that PIDs may now be optimized instead of tuned.

A second classical controller structure, related to the fundamental work of Kalman in the 1960s, is the observer-based controller, which in state-space has the form:

$$K_{obs}(\kappa): \left[\begin{array}{c|c} A + B_2 K_c + K_f C_2 & -K_f \\ \hline K_c & 0 \end{array} \right] \tag{9}$$

Here, the vector of tunable parameters κ regroups the elements of the Kalman gain matrix K_f and the state-feedback control matrix K_c. That is, $\kappa = \left(\mathrm{vec}\left(K_f\right), \mathrm{vec}\left(K_c\right)\right)$. Since the plant P has n_p states, n_y outputs and n_u inputs, κ is of dimension $n_p\left(n_y + n_u\right)$, *i.e.*, $n = n_p\left(n_y + n_u\right) < N$, which indicates that the controller is structured, even though $k = n_p$. In fact, formally the structure of observer-based controllers is defined as

$$\mathcal{K}_{obs} = \left\{ K_{obs}(\kappa) : \textit{as in (9)}, \kappa = \left(\mathrm{vec}\left(K_f\right), \mathrm{vec}\left(K_c\right)\right) \in \mathbb{R}^{n_p(n_y+n_u)} \right\}$$

Now, if we use (9) within the framework of (1), we are computing an observer-based H_∞-controller. However, do not observer-based controllers K_{obs} belong to the realm of H_2-control? This is H_∞ control!

Are we mixing things? Yes we are, but for good reasons! If we are attached to the observer-structure, and at the same time appreciate the robustness of H_∞-control, then we should by all means mix things. The result will be a controller $K_{obs}(\kappa^*)$, where K_c^* gives us two gain matrices K_c^* and K_f^*, neither of which is by itself optimal in any sense[2]. In particular, there are no algebraic Riccati equations for K_f^* or K_c^*. Nonetheless, taken together, they are optimal in the sense that

$$\left\| T_{w\to z}\left(P, K_{obs}(\kappa^*)\right) \right\|_\infty \le \left\| T_{w\to z}\left(P, K_{obs}(\kappa)\right) \right\|_\infty$$

[2] The principle of separation of observation and control is no longer valid.

for any other observer-based controller $K_{obs}(\kappa)$ that stabilizes P internally. In particular, observer-based controllers based on AREs would appear on the right hand side, and hence are sub-optimal.

A third basic controller structure are reduced order controllers. More precisely, the order of K is fixed as $k < n_p$. This is the simplest example of a structure, namely

$$\mathcal{K}_k = \left\{ K : K \text{ as in (3) with size}\left(A_K\right) = k \times k \right\}$$

Here, the vector of tunable elements is $\kappa = \left(\mathrm{vec}\left(A_K\right), \mathrm{vec}\left(B_K\right), \mathrm{vec}\left(C_K\right), \mathrm{vec}\left(D_K\right)\right)$ of dimension $n = k^2 + k(n_y + n_u) + n_y n_u$. This is a structure in the spirit of our definition, because it uses fewer degrees of freedom than the full order controller, which has $N = n_p^2 + n_p(n_y + n_u) + n_y n_u$ free places.

Why is it reasonable to call $\mathcal{K}_k$ a structure as soon as $k < n_p$? The reason is that computing reduced fixed-order optimal H_∞-controllers is substantially more complicated than computing the full-order H_∞ controller. In lieu of two coupled Riccati equations, $K^* \in \mathcal{K}_k$ requires four coupled Riccati equations, [8], and the numerical procedures proposed in the 1990s are clearly demanding. In the realm of matrix inequalities the H_∞-problem for reduced-order controllers has also been well-studied. One obtains an LMI in tandem with a rank constraint, a non-convex problem that is equivalent to a BMI.

Controllers with structure arise naturally. That is why the authors pioneered the investigation of structured H_∞-synthesis in the 1990s.

The solution of the H_∞-control problem

A problem that was left open for 30 years may be expected to be difficult. The difficulty in the H_∞-control problems is due to the fact that it is non-convex, and that the objective in (1) is non-smooth. Moreover, there is a third difficulty, which is related to stability in closed-loop.

Non-smooth optimization

Assuming that $K(\kappa)$ is structured with parameter $\kappa \in \mathbb{R}^n$, we write the closed-loop transfer channel $w \to z$ in (4) as

$$T_{w\to z}\left(P, K(\kappa)\right): \left[\begin{array}{c|c} A\left(K(\kappa)\right) & B\left(K(\kappa)\right) \\ \hline C\left(K(\kappa)\right) & D\left(K(\kappa)\right) \end{array} \right]$$

Then, the H_∞-objective function in (1) becomes

$$\begin{aligned} f(\kappa) &:= \left\| T_{w\to z}\left(P, K(\kappa)\right) \right\|_\infty \\ &= \max_{\omega\in\mathbb{R}} \bar{\sigma}\left(C\left(K(\kappa)\right)\left(j\omega I - A\left(K(\kappa)\right)\right)^{-1} B\left(K(\kappa)\right) + D\left(K(\kappa)\right) \right) \end{aligned} \tag{10}$$

a non-smooth, non-convex function, which in addition is not defined everywhere. Its domain $D_f = \left\{ \kappa \in \mathbb{R}^n : f(\kappa) < \infty \right\}$ contains the internally stabilizing set

$$\begin{aligned} D_s &= \left\{ \kappa \in \mathbb{R}^n : K(\kappa) \text{stabilizes } P \text{internally} \right\} \\ &= \left\{ \kappa \in \mathbb{R}^n : A\left(K(\kappa)\right) \text{ stable} \right\} \end{aligned} \tag{11}$$

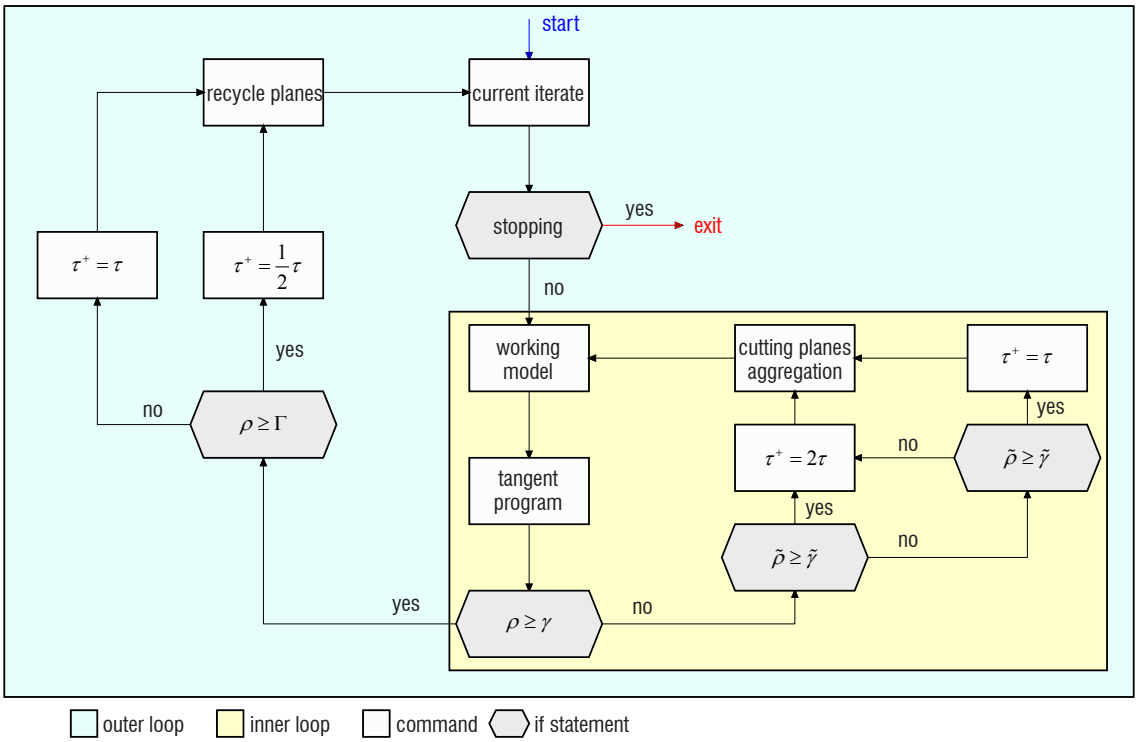

Figure 3 – Flowchart of the proximity control algorithm

The first major step toward the solution of the H_∞ control problem in the seminal paper [2] was to characterize and compute the Clarke subdifferential of the function f. This allowed necessary optimality conditions to be formulated, and thereby enabled locally optimal solutions of (1) to be characterized. These conditions are of primal-dual type, which means that they are expressed in terms of primal variables κ and dual variables X, Y. The latter correspond to the Lyapunov variables used in the ARE and LMI solutions.

The classical solution of the H_∞-problem within $\mathcal{K}_{full}$ using AREs or LMIs for two Lyapunov matrix variables X, Y has the following particularity. The Lyapunov matrices X, Y can be interpreted as the dual variables of our own more general approach, while the primal variable, $K = (A_K, B_K, C_K, D_K)$, can be eliminated. Only in this very specific case is the problem convex in (X, Y).

The second major challenge was to find algorithmic tools to compute solutions of the structured H_∞-problem (1). The objective being non-convex and non-smooth, we had to develop new optimization methods and to prove their convergence. This was started in [2], and continued in [10, 11, 12, 14, 15, 16]. We invented non-convex bundle methods. The bundle technique originated in the 1980s and is the most successful approach to deal with *convex* non-smooth problems in Lagrangian relaxation or stochastic control. We succeeded in extending this to non-convex functions, which represents a major breakthrough.

Stabilization

As we stressed before, the objective $f(\kappa)$ in (1), respectively (10), is only defined on the set

$$D_s = \{\kappa \in \mathbb{R}^n : A(K(\kappa)) \text{is stable}\}$$

from (11). Our optimization method therefore not only has to *iterate* within this set, we first have to *find* a feasible parameter $\kappa \in D_s$. Surprisingly, this is already the first difficulty.

Note that we have to answer the following yes-or-no question:

$$\text{Does there exist } \kappa \text{ such that } A(K(\kappa)) \text{ is stable ?} \qquad (12)$$

Or in our previous notation: Given a controller space $\mathcal{K}$

$$\text{Does there exist } K \in \mathcal{K} \text{ such that } A(K) \text{ is stable ?} \qquad (13)$$

We want an algorithm that computes such a $K \in \mathcal{K}$ if the answer to (13) is "yes", and provides a certificate of non-existence if the answer is "no". Also, we would like these answers reasonably fast, for example, in polynomial time.

How is this related to Kalman's classical theory of stabilizability, detectability, controllability and observability? Stabilizability of (A, B) means that we can stabilize by state feedback. And detectability of (A, C) means that we can add an observer. Therefore, if (A, B) is stabilizable and (A, C) is detectable, then the answer to Question (12) is "yes" for the class $\mathcal{K}_{obs}$ of observer-based controllers. Since stabilizability of (A, B) and detectability of (A, C) are conditions that can be checked by linear algebra (in polynomial time), we can say that (12) is conveniently decided for the class of observer-based controllers $\mathcal{K}_{obs}$ and for any larger class $\mathcal{K} \supset \mathcal{K}_{obs}$.

However, and this is the bad part of the message, for practically important controller structures $K(\kappa)$ the decision (12) is NP-complete. Blondel and Tsitsiklis [17] prove NP-completeness for the classes $\mathcal{K}_k$ of reduced-order controllers, $k < n_p$, including the class $\mathcal{K}_{stat}$ of static controllers, and for the class $\mathcal{K}_{dec}$ of decentralized controllers. It is also known that the decision is hard for PID control. For short, the

most important classes in practice lead already to a difficult problem when it comes to mere stabilization.

Deciding whether a stabilizing controller $K(\kappa)$ with a given structure exists is in general NP-complete.

What does this mean in practice? Complexity theory usually produces pessimistic results. The situation is by no means hopeless. Practical systems are designed to be stabilizable, so as a rule there is a good chance of finding a stabilizing structured controller $K \in \mathcal{K}$ if there is one. What we expect to be hard, is a certificate of non-existence when no such controller exists, because this requires an exhaustive search. Complexity also tells us that we cannot expect a linear algebra procedure as in Kalman's classical theory, at least not one with polynomial complexity. We also know that for most classes $\mathcal{K}$ Problem (12) is decidable, but in exponential time. This follows, for instance, as soon as the problem can be transformed into a polynomial decision problem, to which the Tarski-Seidenberg procedure can, at least in principle, be applied.

Local versus global optimization

The fact that program (1) is non-convex for practical controller structures $\mathcal{K}$ creates a dilemma. Should we go for a globally optimal solution, or should we be modest and be content with locally optimal solutions? In our approach, we have opted for the local approach, since it is more realistic. This does not mean that we advise against the use of global optimization techniques. Such techniques might prove successful for small to medium size problems.

There is, however, one specific global approach on which we wish to comment, because it has contributed substantially to the field of mathematical poppycock. We are speaking about the so-called sums-of-squares (SOS) approach, which is still rumored to be suited for control problems like (1). We now argue that this is a red herring.

For most controller structures $\mathcal{K}$ it is possible to transform program (1) into a bilinear matrix inequality (BMI) using the bounded real lemma. Typically, the BMI is of the form

$$min\left\{c^{\mathrm{T}}x : B(x) \preceq 0\right\} \tag{14}$$

where x now stands for the triple (κ, X, Y) featuring controller gains κ and Lyapunov variables X, Y as unknowns with possibly additional slack variables γ, etc. The SOS approach interprets (14) as a system of polynomial inequalities and uses the sums-of-squares approximation of positive polynomials to creates a hierarchy of LMI problems

$$min\left\{c^{\mathrm{T}}x : L_i(x) \preceq 0\right\} \tag{15}$$

with the property that the solution of (15) converges to the solution of (14). It may even happen that convergence is *finite*, meaning that there exists $i = i(B)$ such that the solution of $min\left\{c^{\mathrm{T}}x : L_{i(B)} \preceq 0\right\}$ solves $min\left\{c^{\mathrm{T}}x : B \preceq 0\right\}$ globally. The way in which this hierarchy is constructed has been much inspired on the idea of a cutting plane proof for a linear integer feasibility problem $Ax \le b,\ x \in \mathbb{Z}^n$.

Let us for simplicity assume that convergence is indeed finite. Then we might be able, it seems, to write down an explicit linear matrix equality

$$min\left\{c^{\mathrm{T}}x : L_{i(B)}(x) \preceq 0\right\} \tag{16}$$

which when solved gives a *globally optimal solution* of (1). (Strictly speaking, we might not be able to write down (16) directly, but rather only to reach it eventually by climbing up in the hierarchy until we get to $i(B)$. This would, of course, spoil the whole idea. However, let us assume, as is often claimed in the SOS community, that we *can* write down (16) explicitly!

Doesn't this sound nice? After all, we have been told since the early 1990s that LMIs can be solved efficiently in quasi-polynomial time. Therefore, all that we have to do is to solve (16) quickly and obtain the global minimum of (14), and respectively of (1).

Of course, this is all rubbish. We *know* that solving Problem (1) globally is NP-complete. The SOS algorithm is even provably exponential. The size of $L_{i(B)} \preceq 0$ grows therefore exponentially in the data size (B). In fact, these problems explode extremely fast. We will need exponential space even to write down $L_{i(B)} \preceq 0$. For sizable plants we might not even be able to *store* the problem on the computer, let alone solve it. The fact that LMIs are solved in polynomial time is pointless, because we are speaking about a problem of *polynomial (exponential)* size.

However, could not something similar be said about *every* global method? Are we too severe when we call SOS a red herring? Indeed, the problem being NP-complete, *every* global method is bound to be exponential. The point is that SOS is a particularly ungainly global method, because it commits two errors, which other global methods may avoid.

The first error is that it transforms (1) to a BMI. This adds a large number of additional variables X, Y, which can be avoided, for example, by our non-smooth approach. We have demonstrated abundantly since the late 1990s that the presence of Lyapunov variables leads to serious ill-conditioning. To wit:

The power oscillation damping control problem, which we solved in [18] using non-smooth optimization, has a system with 90 states, 3 performance connections, 1 input, 1 output, and a controller of reduced order 8. Therefore $\dim(\kappa) = 81$. Transformed to a BMI, it requires additional $3 \cdot \frac{90 \cdot 91}{2} = 12285$ Lyapunov variables. For the SOS approach this is just the bottom line $i = 1$, where the LMI hierarchy starts. The LMI $L_{i(B)} \preceq 0$ will be of size exponential(12366).

The second error in the SOS approach is that it only seeks global minima. That is, it will *not* find local minima of (1) on its way toward the global. This is infelicitous, because local minima are very helpful. They may allow bounds to be improved in branch-and-bound methods, and they give good practical solutions as a rule. The fact that SOS does not use this information (e.g., to infer where it is in the hierarchy $L_i \preceq 0$) is by itself already suspicious.

The H_2/H_∞-problem is also solved

It became already apparent in the 1-DOF scheme (2) that the L^2-L^2, respectively power-to-power, operator norm is not the only possible measure of smallness in a channel $w \to z$. Consider, for instance, the transfer $T_{n_s \to \tilde{u}}$ from sensor noise n_s to the high frequency part $\tilde{u} = W_u u$ of the control law u. If we model n_s as white noise, then

it makes sense to gauge $n_s \to \tilde{u}$ by the operator norm from white noise at the input toward power at the output. This is the H_2-norm. For a stable transfer operator $G(s)$ the H_2-norm is given as

$$\|G\|_2 = \left(\frac{1}{2}\pi\int_0^\infty \mathrm{Tr}\left(G(j\omega)G^H(j\omega)\right)d\omega\right)^{1/2}$$

which makes it an Euclidean norm in the space of stable transfer matrices. Unlike the H_∞-norm, the H_2-norm is *not* an operator norm in the traditional sense. It becomes one as soon as stochastic signals are considered.

$\|w\|$	$\|z\|$	operator norm $\|T_{w\to z}\|$
energy	energy	H_∞
power	power	H_∞
white noise	power	H_2
Sobolev $W^{\infty,\infty}$	L^∞	worst case response norm
L^∞	L^∞	peak gain
past excitation	system ring	Hankel

In the 1-DOF scheme (2) we might decide to use two different norms. We might assess the tracking error $r \to \tilde{e}$ in the H_∞-norm, and the influence of sensor noise on the control $n_s \to \tilde{u}$ by the H_2-norm. Then, we obtain a mixed H_∞ / H_2-control problem

$$\begin{aligned} &\text{minimize} && \|T_{r\to\tilde{e}}(P,K)\|_\infty \\ &\text{subject to} && \|T_{n_s\to\tilde{u}}(P,K)\|_2 \le \gamma_2 \\ & && K \text{ stabilizes } P \text{ internally} \\ & && K = K(\kappa) \text{ has a fixed structure} \end{aligned} \tag{17}$$

where γ_2 is some threshold limiting the power of $\tilde{u}$ in response to white noise in the input n_s. We may introduce the following more abstract setting. Consider a plant in state-space form

$$P: \quad \begin{bmatrix} \dot{x} \\ \hline z_\infty \\ z_2 \\ y \end{bmatrix} = \left[\begin{array}{c|ccc} A & B_\infty & B_2 & B \\ \hline C_\infty & D_\infty & 0 & D_{\infty u} \\ C_2 & 0 & 0 & D_{2u} \\ C & D_{y\infty} & D_{y2} & 0 \end{array}\right] \begin{bmatrix} x \\ \hline w_\infty \\ w_2 \\ u \end{bmatrix} \tag{18}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ the control, $y \in \mathbb{R}^{n_y}$ the output, and where $w_\infty \to z_\infty$ is the H_∞, $w_2 \to z_2$ the H_2 performance channel. Then the mixed H_2 / H_∞-synthesis problem is the optimization program

$$\begin{aligned} &\text{minimize} && \|T_{w_2\to z_2}(P,K)\|_2 \\ &\text{subject to} && \|T_{w_\infty\to z_\infty}(P,K)\|_\infty \le \gamma_\infty \\ & && K \text{ stabilizes } P \text{ internally} \\ & && K \in \mathcal{K} \end{aligned} \tag{19}$$

where $\mathcal{K}$ is a structured controller space as before, and γ_∞ is a suitable threshold, now for the H_∞-norm in the constraint. Notice that the H_2 / H_∞- and H_∞ / H_2-problems are equivalent under suitable choices of γ_2 and γ_∞.

The mixed $K(\kappa)$-synthesis problem with structured controllers $K(\kappa)$ is a natural extension of H_∞-control. This problem also has a long history. It was posed for the first time by Haddad and Bernstein [19] and by Doyle, Zhou, Bodenheimer [20] in 1989. We solved this problem in 2008 in [21].

Naturally, one may immediately think about other multi-objective extensions of (1). For instance, combining the H_∞-norm with time-domain constraints like in IFT(Iterative Feedback Tuning), or H_∞ / H_∞-control. For the first theme, we refer the reader to our solution presented in [22, 23], while H_∞ / H_∞-control will be addressed in the next section.

The H_∞ / H_∞-control problem and other variants

The H_∞ / H_∞-problem can be seen as a special case of (1). Suppose that we have two plants P_1 and P_2 with performance channels $w_i \to z_i$, $i = 1,2$. Assume that the outputs y_i and inputs u_i into P_i have the same dimension, *i.e.*, $dim(y_1) = dim(y_2)$ and $dim(u_1) = dim(u_2)$. Then, we can connect the same controller $u_i = K(\kappa)y_i$ to both plants simultaneously. That is, we may solve a program of the form

$$\begin{aligned} &\text{minimize} && \|T_{w_1\to z_1}(P_1,K)\|_\infty \\ &\text{subject to} && \|T_{w_2\to z_2}(P_2,K)\|_\infty \le \gamma_2 \\ & && K \text{ stabilizes } P_1 \text{ and } P_2 \\ & && K = K(\kappa) \text{ is structured} \end{aligned} \tag{20}$$

It turns out that we may transform (20) favorably into a program of the form

$$\begin{aligned} &\text{minimize} && max\left\{\|T_{w_1\to z_1}(P_1,K(\kappa))\|_\infty, \beta\|T_{w_2\to z_2}(P_2,K(\kappa))\|_\infty\right\} \\ &\text{subject to} && K(\kappa) \text{ stabilizes } P_1 \text{ and } P_2 \end{aligned} \tag{21}$$

which is sometimes called a multidisk problem [10]. For suitable choices of γ_2 and β these two programs are equivalent. However, since the maximum of two H_∞-norms is again an H_∞-norm of an augmented plant, we can solve (21) directly via (1) with a new specific structure, which consists in repeating $K(\kappa)$, as illustrated in Figure 4.

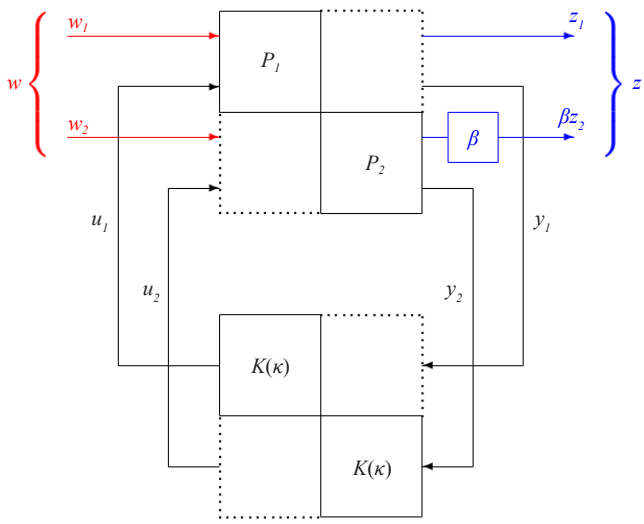

Figure 4 – Illustration of the multidisk problem

and the only connection between the two diagonal parts is the fact that the diagonal block of K is repeated. The objective of (21) is then the channel $w=(w_1,w_2)\to z=(z_1,\beta z_2)$ of the augmented plant. We may now have to update β in order to solve the problem of Figure 4 for a specific γ_2.

Multi-model H_∞-synthesis

Controlling several plants via the same structured controller is one of the most fundamental properties of `hinfstruct` and `systune`. The theoretical basis and implementation of this option has been given in the paper Apkarian and Noll [10] in 2006. The problem setting is as follows. Given N plants $P_1,\ldots,P_N$

$$P_i(s):\left[\begin{array}{c|cc} A_i & B_{1i} & B_{2i} \\ \hline C_{1i} & D_{11i} & D_{12i} \\ C_{2i} & D_{21i} & D_{22i} \end{array}\right], i=1,\ldots,N \tag{22}$$

with identical dimensions of control inputs, that is, $dim(u_1)=\ldots=dim(u_N)$ and also identical dimensions of the measured outputs, *i.e.*, $dim(y_1)=\ldots=dim(y_N)$, we connect the same controller $K(s)$ to these N plants simultaneously. This controller may, in addition, be structured

$$u_i(s)=K(\kappa,s)\,y_i(s), i=1,\ldots,N$$

where κ represents the tunable elements in K. The multi-objective or multidisk H_∞-control problem can now take on several forms, all familiar in multi-objective optimization. For instance, we could consider:

$$\begin{array}{ll} \text{find} & \kappa\in\mathbb{R}^n \\ \text{such that} & \left\|\mathcal{F}_l(P_i,K(\kappa))\right\|_\infty \le \gamma_i,\ i=1,\ldots,N \\ & K(\kappa) \text{ stabilizes } P_i \text{ internally}, i=1,\ldots,N. \end{array} \tag{23}$$

A related form of the multi-objective approach is

$$\begin{array}{ll} \text{minimize} & \left\|\mathcal{F}_l(P_1,K(\kappa))\right\|_\infty \\ \text{such that} & \left\|\mathcal{F}_l(P_i,K(\kappa))\right\|_\infty \le \gamma_i,\ i=2,\ldots,N \\ & K(\kappa) \text{ stabilizes } P_i \text{ internally}, i=1,\ldots,N. \end{array} \tag{24}$$

and the following third form is known as the multidisk approach used in Apkarian and Noll [10]:

$$\min_{\kappa\in\mathbb{R}^n}\ \max_{i=1,\ldots,N}\ \alpha_i\left\|\mathcal{F}_\ell(P_i,K(\kappa))\right\|_\infty \tag{25}$$

where the $\alpha_i>0$ are appropriate weights. We shall in the sequel discuss several examples to understand this approach in practical situations.

Reliable and fault-tolerant control

Reliable or fault-tolerant control is an application of multi-model H_∞-control. The first occurrence of this approach in the literature where an optimization program of the form (22) is applied is Simões *et al.* [26]. In that reference, control of an F-16 aircraft in nominal mode along with 6 failure modes is discussed. These are the failure of left or right stabilizer, failure of left or right aileron, 75% impairment of the stabilizers, and unspecific failure in one of the redundant controllers. Since in each of the 7 modes one needs to ensure satisfactory performance of the system, 3 performance channels are proposed, leading to a total of 21 scenarios, *i.e.*, $N=21$ in (22).

Simultaneous stabilization is NP-complete even in the case of 3 systems and a full order (unstructured) controller. Not surprisingly, simultaneous H_∞-control will therefore fail every now and then. The functions `hinfstruct` and `systune` just offer good practical chances of solving such problems.

System reduction via non-smooth programming

An idea already put forward in our paper [2] is H_∞-system reduction. Consider a stable system

$$G=\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

with size $(A)=n\times n$. Suppose that n is large and that we want to compute a reduced stable system

$$G_{red}=\left[\begin{array}{c|c} A_{red} & B_{red} \\ \hline C_{red} & D_{red} \end{array}\right]$$

of smaller state dimension size $(A_{red})=k\ll n$ that represents G as accurately as possible. The model matching error is $e=(G-G_{red})w$, and after adding a suitable filter W_e (see Figure 5) we might want to have $w\to z$ small in a suitable norm.

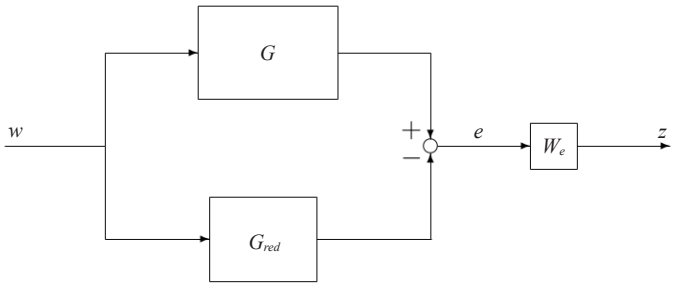

Figure 5 – Illustration of the model reduction problem

The Hankel norm reduction method minimizes $\left\|W_e(G-G_{red})\right\|_H$ in the Hankel norm $\|\cdot\|_H$, the advantage being that the solution can be obtained by linear algebra. A more natural norm would be the H_∞-norm, but the classical balanced reduction method gives only upper bounds of $\left\|W_e(G-G_{red})\right\|_\infty$.

However, we can solve the H_∞-norm reduction problem directly as a special case of (1). In the case $z=e$ without filter we can pass to the standard form by considering the plant

$$P:\left[\begin{array}{c|cc} A & B & 0 \\ \hline C & D & -I \\ 0 & I & 0 \end{array}\right]=\left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right] \tag{26}$$

then G_{red} is the controller, which is of fixed reduced-order.

Note that structured H_∞-filtering is a further application of our non-smooth optimization techniques. A program structure similar to system reduction is obtained in that case.

Optimizing plant and controller simultaneously

In many practical situations it may be advantageous to choose not only the controller, but also certain parameters of the open-loop system in such a way that the entire controlled system behaves optimally in closed loop. Current practice does *not* work this way! What is usually done is that the open-loop system is designed first, and then a feedback controller is computed[3]. However, why not optimize both simultaneously, *i.e.*, plant and controller together? Well, there is now a method that allows this to be done. We illustrate it by means of a simple example.

Consider the second-order spring model with stiffness k, friction f, and mass m

$$P_1 : \begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{k}{m}x_1 - \frac{f}{m}x_2 + \frac{1}{m}w + \frac{1}{m}u \\ y &= x_1 \\ z &= x_1 \end{cases}$$

where w is a disturbance, u is the control, x_1 is the position, and x_2 is the velocity. We measure the position, $y = x_1$, and the position is also the coordinate that we wish to control, hence $z = x_1$. We need a SISO controller $u(s) = K(s)y(s)$ to control the system.

Now, suppose that we also wish to optimize the stiffness k and friction f so that the closed loop system is optimal in the channel $w \to z$. Suppose that we have bounds $\underline{k} \le k \le \overline{k}$ and $\underline{f} \le f \le \overline{f}$ between which we can choose the parameters. And at the same time we want to optimize $K = K(\kappa)$, which can as usual be structured, $\kappa \in \mathbb{R}^n$. Mathematically, this problem can be written as

$$\begin{array}{ll} \text{minimize} & \left\| T_{w\to z}\left(P(k,f), K(\kappa)\right)\right\|_\infty \\ \text{subject to} & K(\kappa) \text{ closed-loop stabilizing} \\ & \underline{k} \le k \le \overline{k}, \underline{f} \le f \le \overline{f} \\ & (k,f,\kappa) \in \mathbb{R}^{n+2} \end{array} \tag{27}$$

Naturally, the non-smooth algorithms in [2] or [14] are versatile and can include such simple bounds easily. They just need expressions for subgradients of the objective with respect to all unknowns (k,f,κ). The question is how this non-standard option is put to work using our non-smooth optimization techniques. The idea is to shuffle all unknown parameters into an augmented structured controller $\widetilde{K}(k,f,\kappa)$, and to connect it to an artificial plant $\widetilde{P}$. The new formulation for (27) is then

$$\begin{array}{ll} \text{minimize} & \left\| T_{w\to z}\left(\widetilde{P}, \widetilde{K}(k,f,\kappa)\right)\right\|_\infty \\ \text{subject to} & \widetilde{K}(k,f,\kappa) \text{ closed-loop stabilizing for } \widetilde{P} \\ & \underline{k} \le k \le \overline{k}, \underline{f} \le f \le \overline{f} \\ & (k,f,\kappa) \in \mathbb{R}^{n+2} \end{array} \tag{28}$$

Note that the routines `hinfstruct` and `systune` automatize this operation so that it remains hidden from the user.

Nonstandard use of H_∞/H_∞-synthesis

The standard way to use multiple H_∞ criteria is certainly in H_∞-loop-shaping, and the documentation of `hinfstruct` makes this a strong point. However, there are some less obvious ideas in which one can use a program of the form (20). Two heuristics for parametric robust control, which we proposed in [24] and [25], can indeed be solved via `hinfstruct` and `systune`.

Control of nonlinear systems with structured H_∞-synthesis

In this section, we discuss a somewhat unexpected application of structured H_∞-synthesis in the control of nonlinear systems. The class of systems that we have in mind are of the form

$$P(y): \begin{array}{l} \dot{x} = A(y)x + B_1(y)w + B_2(y)u \\ z = C_1(y)x + D_{11}(y)w + D_{12}(y)u \\ y = C_2(y)x + D_{21}(y)w + D_{22}(y)u \end{array} \tag{29}$$

where the system matrices depend smoothly on the measured output y. It appears therefore natural to devise a controller of the form

$$K(y): \begin{array}{rcl} \dot{x}_K &=& A_K(y)x_K + B_K(y)y \\ u &=& C_K(y)x_K + D_K(y)y \end{array} \tag{30}$$

which uses the same measurement y available in real time. A natural idea, going back to [29], is to consider y as a time-varying external parameter p and pre-compute $K(p)$ for $P(p)$ for a large set $p \in \Pi$ of possible parameter values. In flight control, for instance, Π is the *flight envelope*, $p = (h,V) \in \mathbb{R}^2$, indexed by altitude h and ground speed V, or sometimes by Mach number and dynamic pressure.

We now propose the following control strategy. In a first step, we pre-compute the optimal H_∞ controller $K^*(p)$ for every $p \in \Pi$ using Program (1):

$$\begin{array}{ll} \text{minimize} & \| T_{w\to z}\left(P(p), K\right)\|_\infty \\ \text{subject to} & K \text{ stabilizes } P(p) \text{ internally} \\ & K \in \mathcal{K} \end{array} \tag{31}$$

The solution $K^*(p)$ of (31) has the structure $\mathcal{K}$. In the terminology of [29], this is the best way to control the system $P(p)$ frozen at $p(t) = y(t)$ instantaneously. In other words, at instant t, we apply the control law $K^*(y(t))$ based on the real-time measurement $y(t)$.

If we could do real-time structured H_∞-synthesis, then controller $K^*(y(t))$ would be computed and applied instantaneously at time t using (31) *and the measurement $y(t)$ available at instant t. As long as this is impossible, we may pre-compute $K^*(p)$ for a large set of possible parameter values $p \in \Pi$, and as soon as $y(t)$ becomes available at time t, look $K^*(y(t))$ up in the table $\{K^*(p) : p \in \Pi\}$, and apply it instantaneously.*

There are two limitations to this ideal approach. Firstly, the ideal table $\{K^*(p) : p \in \Pi\}$ may be too large. And secondly, the behavior of $K^*(p)$ as a function of p may be quite irregular. In fact, it was the latter effect that had stopped this idea in the past[4]. With structured

[3] For instance, in optimal sensor or actuator location the usual line is to optimize P alone, for instance, by maximizing the degree of controllability of P. Instead, one should include K from scratch.

[4] When ARE solvers were used to compute H_∞-controllers, the idea of embedding such a solver into the system obviously came to mind. This failed not due to lack of CPU, but due to the highly irregular behavior of $p \mapsto K^*_{full}(p)$.

control laws $K(\kappa)$ the situation is substantially improved, because one uses fewer degrees of freedom in κ.

What we have tested in [30] is a compromise between optimality of $K^*(p)$ in the sense of Program (31), the necessity to avoid irregular behavior of the curves $p \mapsto K^*(p)$, and the storage requirement of such a law. We use the following definition. A controller parameterization $p \mapsto K(p)$ of the given structure $\mathcal{K}$ is *admissible* for the control of $P(y)$ if the following holds: $K(p)$ stabilizes $P(p)$ internally for every $p \in \Pi$, and

$$\left\| T_{w\to z}\left(P(p), K(p)\right)\right\|_\infty \le (1+\alpha)\left\| T_{w\to z}\left(P(p), K^*(p)\right)\right\|_\infty \quad (32)$$

for every $p \in \Pi$, where α is some fixed threshold, say $\alpha = 0.1\%$. We now seek a parameterization $K(p)$ that is close to the ideal H_∞-parameterization $K^*(p)$ in the sense that (32) is respected, but otherwise is easy to store (to embed) and shows as regular a behavior as possible. Note that (32) allows $K(p)$ to lag behind $K^*(p)$ in performance by no more than $\alpha\%$. Also, observe that this approach is heuristic in so far as internal stability at every $p \in \Pi$ does not guarantee stability of the parameter-varying system as a whole.

Parametric robust H_∞-control

The design of feedback controllers that are robust in the presence of system uncertainty is a recurrent problem in control engineering, from which designers rarely escape due to the inevitable mismatch between a physical system and its mathematical model. It is generally agreed that one should account for the uncertainty already at the modeling stage. In the following, we briefly comment on two such forms of uncertainty: real uncertain parameters Δ_p in the model equations, and complex dynamic uncertainty Δ_d.

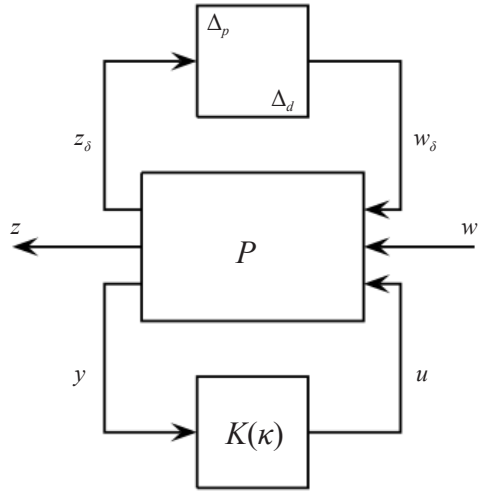

Figure 6 – Illustration of the robust control design problem

Within the H_∞-framework, this *mixed parametric control problem* can be cast as a semi-infinite minmax optimization problem of the form

$$\min_{\kappa\in\mathbb{R}^n} \max_{\Delta\in\mathbf{\Delta}} \left\| T_{w\to z}(\Delta,\kappa)\right\|_\infty \quad (33)$$

where optimization is over a structured control law $K(\kappa)$, as before, but where now in contrast with (25) an infinity of plants, $P(\Delta)$, indexed over a set $\Delta := (\Delta_p, \Delta_d) \in \mathbf{\Delta}$ of mixed uncertain scenarios, has to be controlled *simultaneously*. This problem is therefore considerably more complex than the nominal H_∞-problem (1). Parametric mixed control has been on the control engineering agenda since the late 1970s and 1980s, but no satisfactory solution had been presented until recently. In two recent contributions [9, 38] we have developed a satisfactory mathematically sound solution to this problem, which in parts is already seized by The MathWorks in its 2015b version of the Robust Control Toolbox ■

References

[1] G. ZAMES - *Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverses*. IEEE Transactions on Automatic Control, vol. AC-26, p. 301-320, 1981.

[2] P. APKARIAN, D. NOLL - *Non-Smooth H_∞-Synthesis*. IEEE Transactions on Automatic Control, vol. 51, no. 1, p. 229-244, 2006.

[3] J. V. BURKE, D. HENRION, A. S. LEWIS, M. L. OVERTON - *HIFOO – A Matlab Package for Fixed-Order Controller Design and H_∞ Optimization*. 5th IFAC Symposium on Robust Control Design, Toulouse, France, July 2006.

[4] G. ZAMES - *On the Input-Output Stability of Nonlinear Time-Varying Feedback Systems, Part I and II*. IEEE Transactions on Automatic Control, vol. AC-11, p. 228 and 465, 1966.

[5] J.C. DOYLE, K. GLOVER, P. P. KHARGONEKAR, B. A. FRANCIS - *State-Space Solutions to Standards H_2 and H_∞ Control Problems*. IEEE Transactions on Automatic Control, vol. AC-34, no. 8, p. 831-847, 1989.

[6] J. C. DOYLE - *Robust and Optimal Control*. Proceedings of the 35th Conference on Decision and Control, Kobe, Japan, Conference Proceedings, p. 1595- 1598, 1996.

[7] P. GAHINET, P. APKARIAN - *A Linear Matrix Inequality Approach to H_∞ Control*. International Journal of Robust and Nonlinear Control, 4:421-448, 1994.

[8] D.C. HYLAND, D.S. BERNSTEIN - *The Optimal Projection Equations for Fixed-Order Dynamic Compensation*. IEEE Trans. Autom. Control, vol. AC-29, no. 11, p. 1034-1037, 1984.

[9] P. APKARIAN, M. N. DAO, D. NOLL - *Parametric Robust Structured Control Design*. IEEE Transactions on Automatic Control, vol. 60, issue 7, p. 1857-1869, 2015.

[10] P. APKARIAN, D. NOLL - *Non-Smooth Optimization for Multidisk H_∞ Synthesis*. European Journal of Control, vol. 12, no. 3, p. 229-244, 2006.

[11] P. APKARIAN, D. NOLL - *Non-Smooth Optimization for Multiband Frequency Domain Control Design*. Automatica, vol. 43, no. 4, p. 724-731, 2007.

[12] P. APKARIAN, D. NOLL, O. PROT - *A Trust Region Spectral Bundle Method for Non-Convex Eigenvalue Optimization*. SIAM Journal on Optimization, vol. 19, no. 1, p. 281-306, 2008.

[13] P. APKARIAN, D. NOLL, L. RAVANBOD - *Branch and Bound Algorithm with Applications to Robust Stability*. Journal of Global Optimization, published electronically march 2016. DOI 10.1007/s10898-016-0424-6.

[14] D. NOLL, O. PROT, A. RONDEPIERRE - *A Proximity Control Algorithm to Minimize Non-Smooth and Non-Convex Functions*. Pacific Journal of Optimization, vol. 4, no. 3, pp. 569-602, 2008.

[15] P. APKARIAN, D. NOLL, O. PROT - *A Proximity Control Algorithm to Minimize Non-Smooth and Non-Convex Semi-Infinite Maximum Eigenvalue Functions*. Journal of Convex Analysis, vol. 16, no. 3 & 4, p. 641-666, 2009.

[16] D. NOLL - *Cutting Plane Oracles to Minimize Non-Smooth and Non-Convex Functions*. Journal of Set-Valued and Variational Analysis, vol. 18, no. 3-4, p. 531-568, 2010.

[17] V. D. BLONDEL, J. N. TSITSIKLIS - *NP-Hardness of Some Linear Control Design Problems*. SIAM J. of Control and Opt., 35:6, p. 2118-2127, 1997.

[18] P. APKARIAN, A. SIMOES, D. NOLL - *A Non-Smooth Progress Function Algorithm for Frequency Shaping Control Design*. IET Control Theory & Applications, vol. 2, no. 4, pp. 323 - 336, 2008.

[19] W. M. HADDAD, D. S. BERNSTEIN - *LQG Control with a H_∞ Performance Bound: A Riccati Equation Approach*. IEEE Trans. Aut. Control, AC-34(3):293-305, 1989.

[20] J. C. DOYLE, K. ZHOU, B. BODENHEIMER - *Optimal Control with Mixed H_2 and H_∞ Performance Objectives*. Proceedings of the American Control Conference, vol.3, p. 2065-2070, 1989.

[21] P. APKARIAN, D. NOLL, A. RONDEPIERRE - *Mixed H_2 / H_∞ Control via Nonsmooth Optimization*. SIAM Journal on Control and Optimization, vol. 47, no. 3, p. 1516-1546, 2008.

[22] P. APKARIAN, L. RAVANBOD, D. NOLL - *Time Domain Constrained Structured H_∞ Synthesis*. International Journal of Robust and Nonlinear Control, vol. 21, no. 2, p. 197-217, 2011.

[23] P. APKARIAN, D. NOLL, A. SIMÕES - *Time Domain Control Design: A Non-Smooth Approach*. IEEE Transactions on Control Systems Technology, vol. 17, no. 6, p. 1439-1445, 2009.

[24] L. RAVANBOD, D. NOLL, P. APKARIAN - *Robustness via Structured H_∞ / H_∞-Synthesis*. International Journal of Control, vol. 84, issue 5, 2011, p. 851-866, 2011.

[25] P. APKARIAN - *Non-Smooth μ-Synthesis*. International Journal of Robust and Nonlinear Control, vol. 21(8), p. 1493-1508, 2011.

[26] P. APKARIAN, A. SIMÕES, D. NOLL - *Non-Smooth Multi-Objective Synthesis with Applications*. Control Engineering Practice, 17(11):2009,1338-1348.

[27] L. RAVANBOD, D. NOLL - *Gain-Scheduled Autopilot for an Aircraft*. 7th IFAC Symposium on Robust Control Design, 2012 Aalborg. Conference Proceedings, 2011.

[28] 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), Nara, Japan, September 6-10, 2010.

[29] J. S. SHAMMA, M. ATHANS - *Analysis of Gain Scheduled Control for Nonlinear Plants*. IEEE Transactions on Automatic Control, vol. 35, no. 4, p. 898-907, 1990.

[30] L. RAVANBOD, D. NOLL - *Gain-Scheduled PID for Imbalance Compensation of a Magnetic Bearing*. 8th International Conference on Informatics, Automation and Robotics (ICINCO 2011), Noordwijkerhout, The Netherlands, Conference Proceedings, p. 330-337, July 2011.

[31] S. SKOGESTAD, I. POSTLETHWAITE - *Multivariable Feedback Control*. Wiley, 2005.

[32] N. ABROUG, B. MORINIERE - *Enhancing Motor Torque Control by Implementing H_∞ Controller and Compensating Electronics Nonlinearities*. IEEE 23rd International Symposium on Industrial Electronics, p. 111-116, 2014.

[33] C. PITTET, P. PRIEUR - *Structured Accelero-Stellar Estimator for Microscope Drag-Free Mission*. Advances in Aerospace Guidance, Navigation and Control, p. 591-604, 2015.

[34] S. HIRWA, P. FEYEL, G. DUC, G. SANDOU - *Reduced-Order Controllers for Line of Sight Stabilization*. 7th IFAC Symposium on Robust Control Design, vol. 45, no. 13, p. 103-108, 2012.

[35] A. FALCOZ, P. CHRISTELLE, S. BENNANI, A. GUIGNARD, B. FRAPARD - *Systematic Design Methods of Robust and Structured Controllers for Satellites*. CEAS Space Journal, vol. 7, no. 3, p. 319-334; 2015.

[36] C. MEYER, J. PRODIGUE, G. BROUX, O. CANTINAUD, C. POUSSOT-VASSAL - *Ground Test for Vibration Control Demonstrator*. International Conference on Motion and Vibration Control (MOVIC'16), 2016.

[37] C. POUSSOT-VASSAL, P. VUILLEMIN - *Introduction to MORE: A MOdel REduction Toolbox.* IEEE Multi-Conference on Systems and Control, p. 776-781, 2012.

[38] R. S. DE AGUIAR, P. APKARIAN, D.NOLL - *Structured Robust Control against Mixed Uncertainty.* Submitted.

AUTHORS

Pierre Apkarian received the Ph.D. degree in control engineering from the *Ecole Nationale Supérieure de l'Aéronautique et de l'Espace* (ENSAE), France, in 1988. He was qualified as a Professor from University of Toulouse (France) in both control engineering and applied mathematics in 1999 and 2001, respectively. Since 1988, he has been a Research Scientist at ONERA (*Office National d'Etudes et de Recherches Aérospatiales*) and an Associate Professor at the University of Toulouse. Pierre Apkarian has served as an associate editor for the IEEE Transactions on Automatic Control. His research interests include robust and gain-scheduling control theory through LMI methods or optimization-based techniques. More recently, his research has focused on the development of specialized non-smooth programming techniques for control system design. He is co-author of the `hinfstruct` and `systune` software in MATLAB's Robust Control Toolbox.

Dominikus Noll received his Ph.D. and habilitation in 1983 and 1989 from *Universität Stuttgart* (Germany). Since 1995 he is a professor of applied mathematics at the University of Toulouse (France), and a distinguished professor of mathematics since 2009. Dr. Noll has held visiting positions at Uppsala University, Dalhousie University, the University of Waterloo, Simon Fraser University, and the University of British Columbia. Dr. Noll's current research interests include nonlinear optimization, optimal control, projection-based iterative schemes, and robust feedback control design. He is co-inventor of the synthesis tools `hinfstruct` and `systune`. Dr. Noll is Associate Editor of the Journal of Convex Analysis.