Optimization-Based Control Design Techniques and Tools

Pierre Apkarian[1] and Dominikus Noll[2]
[1]Control Department, DCSD, ONERA – The French Aerospace Lab, Toulouse, France
[2]Institut de Mathématiques, Université de Toulouse, Toulouse, France

Abstract

Structured output feedback controller synthesis is an exciting recent concept in modern control design, which bridges between theory and practice in so far as it allows for the first time to apply sophisticated mathematical design paradigms like H_∞- or H_2-control within control architectures preferred by practitioners. The new approach to structured H_∞-control, developed by the authors during the past decade, is rooted in a change of paradigm in the synthesis algorithms. Structured design is no longer based on solving algebraic Riccati equations or matrix inequalities. Instead, optimization-based design techniques are required. In this essay we indicate why structured controller synthesis is central in modern control engineering. We explain why non-smooth optimization techniques are needed to compute structured control laws, and we point to software tools which enable practitioners to use these new tools in high-technology applications.

Keywords

Controller tuning · H_∞ synthesis · Multi-objective design · Nonsmooth optimization · Structured controllers · Robust control

Introduction

In the modern high-technology field, control engineers usually face a large variety of concurring design specifications such as noise or gain attenuation in prescribed frequency bands, damping, decoupling, constraints on settling time or risetime, and much else. In addition, as plant models are generally only approximations of the true system dynamics, control laws have to be robust with respect to uncertainty in physical parameters or with regard to un-modeled high-frequency phenomena. Not surprisingly, such a plethora of constraints presents a major challenge for controller tuning, due not only to the ever growing number of such constraints but also because of their very different provenience.

The steady increase in plant complexity is exacerbated by the quest that regulators should be as simple as possible, easy to understand and to tune by practitioners, convenient to hardware implement, and generally available at low cost. These practical constraints highlight the limited use of Riccati- or LMI-based controllers, and

J. Baillieul, T. Samad (eds.), *Encyclopedia of Systems and Control*,
https://doi.org/10.1007/978-1-4471-5102-9_144-2

they are the driving force for the implementation of *structured* control architectures. On the other hand, this means that hand-tuning methods have to be replaced by rigorous algorithmic optimization tools.

Structured Controllers

Before addressing specific optimization techniques, we recall some basic terminology for control design problems with structured controllers. The plant model P is described as

$$P : \begin{cases} \dot{x}_P = Ax_P + B_1 w + B_2 u \\ z \ = C_1 x_P + D_{11} w + D_{12} u \\ y \ = C_2 x_P + D_{21} w + D_{22} u \end{cases} \tag{1}$$

where A, B_1, … are real matrices of appropriate dimensions, $x_P \in \mathbb{R}^{n_P}$ 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, the sought output feedback controller K is described as

$$K : \begin{cases} \dot{x}_K = A_K x_K + B_K y \\ u = C_K x_K + D_K y \end{cases} \tag{2}$$

with $x_K \in \mathbb{R}^{n_K}$ and is called *structured* if the (real) matrices A_K, B_K, C_K, D_K depend smoothly on a design parameter $\mathbf{x} \in \mathbb{R}^n$, referred to as the vector of tunable parameters. Formally, we have differentiable mappings

$$A_K = A_K(\mathbf{x}), B_K = B_K(\mathbf{x}), C_K = C_K(\mathbf{x}),$$
$$D_K = D_K(\mathbf{x}),$$

and we abbreviate these by the notation $K(\mathbf{x})$ for short to emphasize that the controller is structured with $\mathbf{x}$ as tunable elements. A structured controller synthesis problem is then an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & \|T_{wz}(P, K(\mathbf{x}))\| \\ \text{subject to} & K(\mathbf{x}) \text{ closed-loop stabilizing} \\ & K(\mathbf{x}) \text{ structured, } \mathbf{x} \in \mathbb{R}^n \end{array} \tag{3}$$

where $T_{wz}(P, K) = \mathcal{F}_\ell(P, K)$ is the lower feedback connection of (1) with (2) as in Fig. 1 (left), also called the linear fractional transformation (Zhou et al. 1996). The norm $\|\cdot\|$ stands for the H_∞-norm, the H_2-norm, or any other system norm, while the optimization variable $\mathbf{x} \in \mathbb{R}^n$ regroups the tunable parameters in the design.

Standard examples of structured controllers $K(\mathbf{x})$ include realizable PIDs, observer-based, reduced-order, or decentralized controllers, which in state space are expressed as:

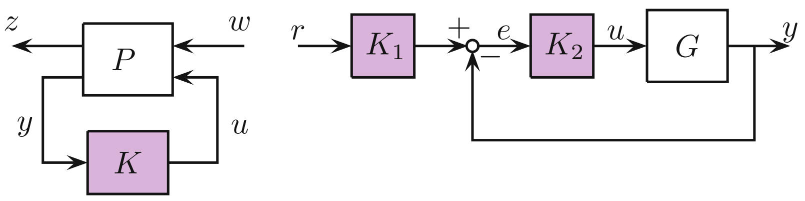
$$\left[\begin{array}{cc|c} 0 & 0 & 1 \\ 0 & -1/\tau & -k_D/\tau \\ \hline k_I & 1/\tau & k_P + k_D/\tau \end{array}\right],$$
$$\left[\begin{array}{c|c} A - B_2 K_c - K_f C_2 & K_f \\ \hline -K_c & 0 \end{array}\right], \quad \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right],$$
$$\left[\begin{array}{c|c} \operatorname{diag}(A_{K_1}, \ldots, A_{K_q}) & \operatorname{diag}(B_{K_1}, \ldots, B_{K_q}) \\ \hline \operatorname{diag}(C_{K_1}, \ldots, C_{K_q}) & \operatorname{diag}(D_{K_1}, \ldots, D_{K_q}) \end{array}\right].$$

In the case of a PID, the tunable parameters are $\mathbf{x} = (\tau, k_P, k_I, k_D)$; for observer-based controllers, $\mathbf{x}$ regroups the estimator and state-feedback gains (K_f, K_c); for reduced order controllers $n_K < n_P$, the tunable parameters $\mathbf{x}$ are the $n_K^2 + n_K n_y + n_K n_u + n_y n_u$ unknown entries in (A_K, B_K, C_K, D_K); and in the decentralized form, $\mathbf{x}$ regroups the unknown entries in $A_{K1}, \ldots, D_{Kq}$. In contrast, full-order controllers have the maximum number $N = n_P^2 + n_P n_y + n_P n_u + n_y n_u$ of degrees of freedom and are referred to as unstructured or as *black-box* controllers.

More sophisticated controller structures $K(\mathbf{x})$ arise from architectures like a 2-DOF control arrangement with feedback block K_2 and a set-point filter K_1 as in Fig. 1 (right). Suppose K_1 is the 1st-order filter $K_1(s) = a/(s + a)$ and K_2 the PI feedback $K_2(s) = k_P + k_I/s$. Then the transfer T_{ry} from r to y can be represented as the feedback connection of P and $K(\mathbf{x}, s)$ with

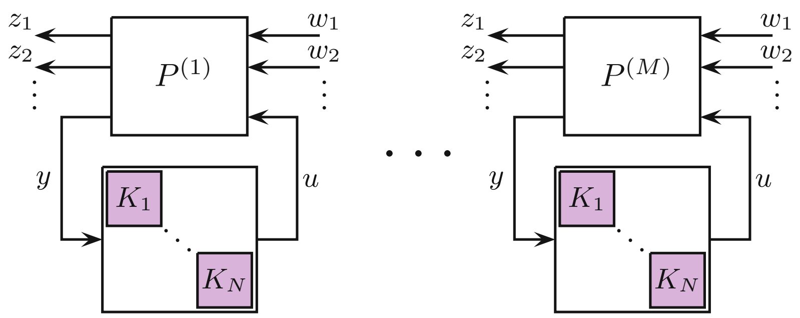
$$P = \left[\begin{array}{c|cc} A & 0 & B \\ \hline C & 0 & D \\ 0 & I & 0 \\ -C & I & -D \end{array}\right],$$

$$K(\mathbf{x}, s) = \left[\, K_1(a, s) \quad K_2(k_P, k_I, s) \,\right],$$



Optimization-Based Control Design Techniques and Tools, Fig. 1 Black-box full-order controller K on the left, structured 2-DOF control architecture with $K = \mathrm{diag}(K_1, K_2)$ on the right

Optimization-Based Control Design Techniques and Tools, Fig. 2 Synthesis of $K = \mathrm{diag}(K_1, \ldots, K_N)$ against multiple requirements or models $P^{(1)}, \ldots, P^{(M)}$. Each $K_i(\mathbf{x})$ can be structured



and gathering tunable elements in $\mathbf{x} = (a, k_P, k_I)$.

In much the same way, arbitrary multi-loop interconnections of fixed-model elements with tunable controller blocks $K_i(\mathbf{x})$ can be rearranged as in Fig. 2, so that $K(\mathbf{x})$ captures all tunable blocks in a decentralized structure general enough to cover most engineering applications.

The structure concept is equally useful to deal with the second central challenge in control design: *system uncertainty*. The latter may be handled with μ-synthesis techniques (Stein and Doyle 1991) if a parametric uncertain model is available. A less ambitious but often more practical alternative consists in optimizing the structured controller $K(\mathbf{x})$ against a finite set of plants $P^{(1)}, \ldots, P^{(M)}$ representing model variations due to uncertainty, aging, sensor and actuator breakdown, un-modeled dynamics, in tandem with the robustness and performance specifications. This is again formally covered by Fig. 2 and leads to a multi-objective constrained optimization problem of the form

$$
\begin{aligned}
&\text{minimize } f(\mathbf{x}) = \max_{k \in \mathrm{SOFT}, i \in I_k} \|T^{(k)}_{w_i z_i}(K(\mathbf{x}))\| \\
&\text{subject to } g(\mathbf{x}) = \max_{k \in \mathrm{HARD}, j \in J_k} \|T^{(k)}_{w_j z_j}(K(\mathbf{x}))\| \le 1 \\
&\qquad K(\mathbf{x}) \text{ structured and stabilizing} \\
&\qquad \mathbf{x} \in \mathbb{R}^n \qquad (4)
\end{aligned}
$$

where $T^{(k)}_{w_i z_i}$ denotes the ith closed-loop robustness or performance channel $w_i \to z_i$ for the k-th plant model $P^{(k)}$. SOFT and HARD denote index sets taken over a finite set of specifications, say in $\{1, \ldots, M\}$. The rationale of (4) is to minimize the worst-case cost of the soft constraints $\|T^{(k)}_{w_i z_i}\|$, $k \in \mathrm{SOFT}$, while enforcing the hard constraints $\|T^{(k)}_{w_j z_j}\| \le 1$, $k \in \mathrm{HARD}$, which prevail over soft ones and are mandatory. In addition to local optimization (4), the problem can undergo a global optimization step in order to prove global stability and performance of the design, see Ravanbod et al. (2017) and Apkarian et al. (2015a, b).

Optimization Techniques Over the Years

During the late 1990s, the necessity to develop design techniques for structured regulators $K(\mathbf{x})$ was recognized (Fares et al. 2001), and the limitations of synthesis methods based on algebraic Riccati equations or linear matrix inequalities (LMIs) became evident, as these techniques cannot provide structured controllers needed in practice. The lack of appropriate synthesis techniques for structured $K(\mathbf{x})$ led to the unfortunate situation, where sophisticated approaches like the H_∞ paradigm developed by academia since the 1980s could not be brought to work for the design of those controller structures $K(\mathbf{x})$ preferred by practitioners. Design engineers had to continue to rely on heuristic and ad hoc tuning techniques, with only limited scope and reliability. As an example, post-processing to reduce a black-box controller to a practical size is prone to failure. It may at best be considered a fill-in for a rigorous design method which directly computes a reduced-order controller. Similarly, hand-tuning of the parameters $\mathbf{x}$ remains a puzzling task because of the loop interactions and fails as soon as complexity increases.

In the late 1990s and early 2000s, a change of methods was observed. Structured H_2- and H_∞-synthesis problems (3) were addressed by bilinear matrix inequality (BMI) optimization, which used local optimization techniques based on the augmented Lagrangian method (Fares et al. 2001; Noll et al. 2004; Kocvara and Stingl 2003; Noll 2007), sequential semidefinite programming methods (Fares et al. 2002; Apkarian et al. 2003), and non-smooth methods for BMIs (Noll et al. 2009; Lemaréchal and Oustry 2000). However, these techniques were based on the bounded real lemma or similar matrix inequalities and were therefore of limited success due to the presence of Lyapunov variables, i.e., matrix-valued unknowns, whose dimension grows quadratically in $n_P + n_K$ and represents the bottleneck of that approach.

The epoch-making change occurs with the introduction of non-smooth optimization techniques (Noll and Apkarian 2005; Apkarian and Noll 2006b, c, 2007) to programs (3) and (4). Today non-smooth methods have superseded matrix inequality-based techniques and may be considered the state of art as far as realistic applications are concerned. The transition took almost a decade.

Alternative control-related local optimization techniques and heuristics include the gradient sampling technique of Burke et al. (2005), and other derivative-free optimization techniques as in Kolda et al. (2003) and Apkarian and Noll (2006a), particle swarm optimization, see Oi et al. (2008) and references therein, and also evolutionary computation techniques (Lieslehto 2001). All these classes do not exploit derivative information and rely on function evaluations only. They are therefore applicable to a broad variety of problems including those where function values arise from complex numerical simulations. The combinatorial nature of these techniques, however, limits their use to small problems with a few tens of variable. More significantly, these methods often lack a solid convergence theory. In contrast, as we have demonstrated over recent years (Apkarian and Noll 2006b; Noll et al. 2008; Apkarian et al. 2016, 2018), specialized non-smooth techniques are highly efficient in practice, are based on a sophisticated convergence theory, are capable of solving medium-size problems in a matter of seconds, and are still operational for large-size problems with several hundreds of states.

Non-smooth Optimization Techniques

The benefit of the non-smooth casts (3) and (4) lies in the possibility to avoid searching for Lyapunov variables, a major advantage as their number $(n_P + n_K)^2/2$ usually largely dominates n, the number of true decision parameters $\mathbf{x}$. Lyapunov variables do still occur implicitly in the function evaluation procedures, but this has no harmful effect for systems up to several hundred states. In abstract terms, a non-smooth optimization program has the form

$$\begin{array}{rl} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & g(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{array} \qquad (5)$$

where $f, g : \mathbb{R}^n \to \mathbb{R}$ are locally Lipschitz functions and are easily identified from the cast in (4).

In the realm of convex optimization, non-smooth programs are conveniently addressed by so-called bundle methods, introduced in the late 1970s by Lemaréchal (1975). Bundle methods are used to solve difficult problems in integer programming or in stochastic optimization via Lagrangian relaxation. Extensions of the bundling technique to non-convex problems like (3) or (4) were first developed in Apkarian and Noll (2006b, c, 2007), Apkarian et al. (2008), and Noll et al. (2009) and, in more abstract form, in Noll et al. (2008). Recently, we also extended bundle techniques to the trust-region framework (Apkarian et al. 2016, 2018; Apkarian and Noll 2018), which leads to the first extension of the classical trust-region method to non-differential optimization supported by a valid convergence theory.

Figure 3 shows a schematic view of a non-convex bundle method consisting of a descent-step generating inner loop (yellow block) comparable to a line search in smooth optimization, embedded into the outer loop (blue box), where serious iterates are processed, stopping criteria are applied, and the acceptance rules of traditional trust-region techniques are assured. At the core of the interaction between inner and outer loop is the management of the proximity control parameter τ, which governs the stepsize $\|\mathbf{x} - \mathbf{y}^k\|$ between trial steps $\mathbf{y}^k$ at the current serious iterate $\mathbf{x}$. Similar to the management of a trust-region radius or of the step size in a line search, proximity control allows to force shorter trial steps if agreement of the local model with the true objective function is poor and allows larger steps if agreement is satisfactory.

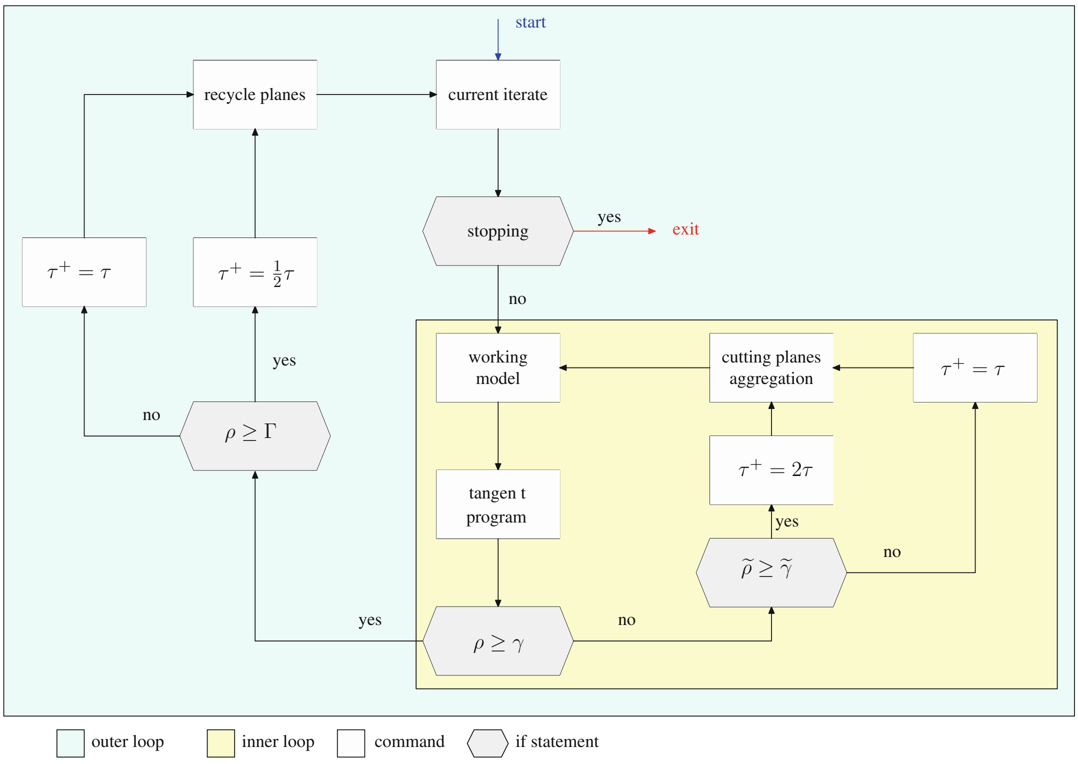
Oracle-based bundle methods traditionally assure global convergence in the sense of subsequences under the sole hypothesis that for every trial point $\mathbf{x}$, the function value $f(\mathbf{x})$ and one Clarke subgradient $\phi \in \partial f(\mathbf{x})$ are provided. In automatic control applications, it is as a rule possible to provide more specific information, which may be exploited to speed up convergence (Apkarian and Noll 2006b).

Computing function value and gradients of the H_2-norm $f(\mathbf{x}) = \|T_{wz}(P, K(\mathbf{x}))\|_2$ requires essentially the solution of two Lyapunov equations of size $n_P + n_K$; see Apkarian et al. (2007). For the H_∞-norm, $f(\mathbf{x}) = \|T_{wz}(P, K(\mathbf{x}))\|_\infty$, function evaluation is based on the Hamiltonian algorithm of Benner et al. (2012) and Boyd et al. (1989). The Hamiltonian matrix is of size $2(n_P + n_K)$, so that function evaluations may be costly for very large plant state dimension ($n_P > 500$), even though the number of outer loop iterations of the bundle algorithm is not affected by a large n_P and generally relates to n, the dimension of $\mathbf{x}$. The additional cost for subgradient computation for large n_P is relatively cheap as it relies on linear algebra (Apkarian and Noll 2006b). Function and subgradient evaluations for H_∞ and H_2 norms are typically obtained in $\mathcal{O}\left((n_P + n_K)^3\right)$ flops.

Computational Tools

Our non-smooth optimization methods became available to the engineering community since 2010 via the MATLAB Robust Control Toolbox (Robust Control Toolbox 4.2 2012; Gahinet and Apkarian 2011). Routines `HINFSTRUCT`, `LOOPTUNE`, and `SYSTUNE` are versatile enough to define and combine tunable blocks $K_i(\mathbf{x})$, to build and aggregate multiple models and design requirements on $T_{wz}^{(k)}$ of different nature, and to provide suitable validation tools. Their implementation was carried out in cooperation with P. Gahinet (MathWorks). These routines further exploit the structure of problem (4) to enhance efficiency; see Apkarian and Noll (2007) and Apkarian and Noll (2006b).

It should be mentioned that design problems with multiple hard constraints are inherently complex and generally NP-hard, so that exhaustive methods fail even for small- to medium-size problems. The principled decision made in



Optimization-Based Control Design Techniques and Tools, Fig. 3 Flow chart of proximity control bundle algorithm

Apkarian and Noll (2006b), and reflected in the MATLAB tools, is to rely on local optimization techniques instead. This leads to weaker convergence certificates but has the advantage to work successfully in practice. In the same vein, in (4) it is preferable to rely on a mixture of soft and hard requirements, for instance, by the use of exact penalty functions (Noll and Apkarian 2005). Key features implemented in the mentioned MATLAB routines are discussed in Apkarian (2013), Gahinet and Apkarian (2011), and Apkarian and Noll (2007).

Applications

Design of a feedback regulator is an interactive process, in which tools like SYSTUNE, LOOPTUNE, or HINFSTRUCT support the designer in various ways. In this section we illustrate their enormous potential by showing that even infinite-dimensional systems may be successfully addressed by these tools. For a plethora of design examples for real-rational systems including parametric and complex dynamic uncertainty, we refer to Ravanbod et al. (2017), Apkarian et al. (2015a, 2016, 2018), and Apkarian and Noll (2018). For recent applications of our tools in real-world applications, see also Falcoz et al. (2015), where it is in particular explained how HINFSTRUCT helped in 2014 to save the Rosetta mission. Another important application of HINFSTRUCT is the design of the atmospheric flight pilot for the ARIANE VI launcher by the ArianeGroup (Ganet-Schoeller et al. 2017).

Illustrative Example

We discuss boundary control of a wave equation with anti-stable damping,

$$\begin{aligned} x_{tt}(\xi,t) &= x_{\xi\xi}(\xi,t), \quad t \geq 0, \xi \in [0,1] \\ x_\xi(0,t) &= -q x_t(0,t), \quad q > 0, q \neq 1 \qquad (6) \\ x_\xi(1,t) &= u(t). \end{aligned}$$

where notations x_y and x_{yy} stand for partial first- and second-order derivatives of x with respect to y, respectively. In (6), $x(\cdot,t), x_t(\cdot,t)$ is the state; the control applied at the boundary $\xi = 1$ is $u(t)$, and we assume that the measured outputs are

$$\begin{aligned} y_1(t) &= x(0,t), y_2(t) = x(1,t), \\ y_3(t) &= x_t(1,t). \qquad (7) \end{aligned}$$

The system has been discussed previously in Smyshlyaev and Krstic (2009), Fridman (2014), and Bresch-Pietri and Krstic (2014) and has been proposed for the control of slip-strick vibrations in drilling devices (Saldivar et al. 2013). Here measurements y_1, y_2 correspond to the angular positions of the drill string at the top and bottom level, and y_3 measures angular speed at the top level, while control corresponds to a reference velocity at the top. The friction characteristics at the bottom level are characterized by the parameter q, and the control objective is to maintain a constant angular velocity at the bottom.

Similar models have been used to control pressure fields in duct combustion dynamics; see DeQueiroz and Rahn (2002). The challenge in (6) and (7) is to design implementable controllers despite the use of an infinite-dimensional system model.

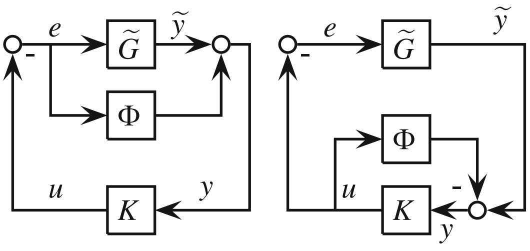
The transfer function of (6) is obtained from

$$G(\xi,s) = \frac{x(\xi,s)}{u(s)} = \frac{1}{s} \cdot \frac{(1-q)e^{s\xi} + (1+q)e^{-s\xi}}{(1-q)e^{s} - (1+q)e^{-s}},$$

which in view of (7) leads to $G(s)^T = [G_1\ G_2\ G_3] = [G(0,s)\ G(1,s)\ sG(1,s)]$.

Putting G in feedback with the controller $K_0 = [0\ 0\ 1]$ leads to $\widehat{G} = G/(1+G_3)$, where

$$\begin{aligned} \widehat{G}(s) &= \begin{bmatrix} \frac{1}{s(1-q)} \\ \frac{1+Q}{2s} \\ \frac{1}{2} \end{bmatrix} + \begin{bmatrix} -\frac{1-e^{-s}}{s(1-q)} \\ -\frac{Q(1-e^{-2s})}{2s} \\ \frac{Q}{2}e^{-2s} \end{bmatrix} \\ &=: \widetilde{G}(s) + \Phi(s), \qquad (8) \end{aligned}$$



Optimization-Based Control Design Techniques and Tools, Fig. 4 Stability of the closed-loop $(\widetilde{G} + \Phi, K)$ is equivalent to stability of the closed-loop $(\widetilde{G}, \texttt{feedback}(K,\Phi))$. See also Moelja and Meinsma (2003)

where $Q = (1+q)/(1-q)$.

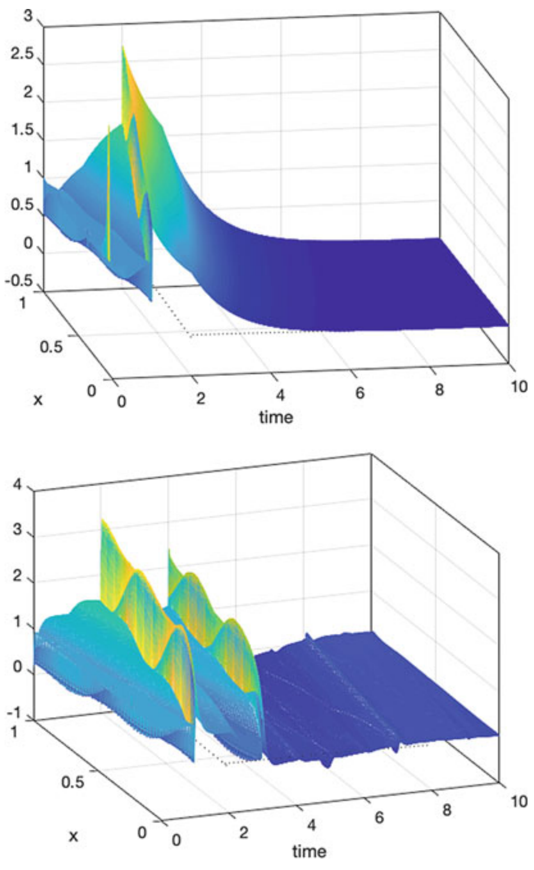
Here $\widetilde{G}$ is real-rational and unstable, while Φ is stable but infinite dimensional. The latter follows from the fact that $\frac{1-e^{-s}}{s}$ and $\frac{1-e^{-2s}}{2s}$ are stable transfert functions as clarified by series expansions. Now we use the fact that stability of the closed loop $(\widetilde{G} + \Phi, K)$ is equivalent to stability of the loop $(\widetilde{G}, \texttt{feedback}(K,\Phi))$ upon defining $\texttt{feedback}(M,N) := M(I+NM)^{-1}$. The loop transformation is explained in Fig. 4.

Using (8) we construct a finite-dimensional structured controller $\widetilde{K} = \widetilde{K}(\mathbf{x})$ which stabilizes $\widetilde{G}$. The controller K stabilizing $\widehat{G}$ in (8) is then recovered from $\widetilde{K}$ through the equation $\widetilde{K} = \texttt{feedback}(K,\Phi)$, which when inverted gives $K = \texttt{feedback}(\widetilde{K},-\Phi)$. The overall controller for (6) is $K^* = K_0 + K$, and since along with K only delays appear in Φ, the controller K^* is implementable.

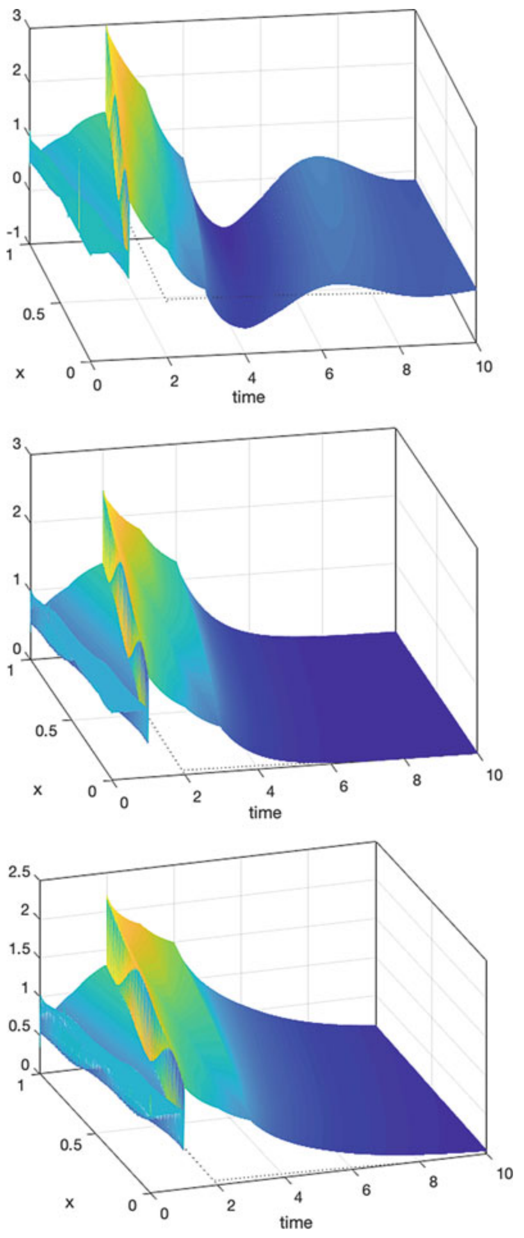
Construction of $\widetilde{K}$ uses SYSTUNE with pole placement via TuningGoal.Poles, imposing that closed-loop poles have a minimum decay of 0.9, minimum damping of 0.9, and a maximum frequency of 4.0. The controller structure is chosen as static, so that $\mathbf{x} \in \mathbb{R}^3$. A simulation with K^* is shown in Fig. 5 (bottom), and some acceleration over the backstepping controller from Bresch-Pietri and Krstic (2014) (top) is observed.

Gain-Scheduling Control

Our last study is when the parameter $q \geq 0$ is uncertain or allowed to vary in time with



Optimization-Based Control Design Techniques and Tools, Fig. 5 Wave equation. Simulations for K obtained by backstepping control (top) (Bresch-Pietri and Krstic 2014) and $K^* = K_0 + K$ obtained by optimizing $\texttt{feedback}(\widetilde{G}, \widetilde{K})$ via SYSTUNE (bottom). Both controllers are ∞-dimensional but implementable

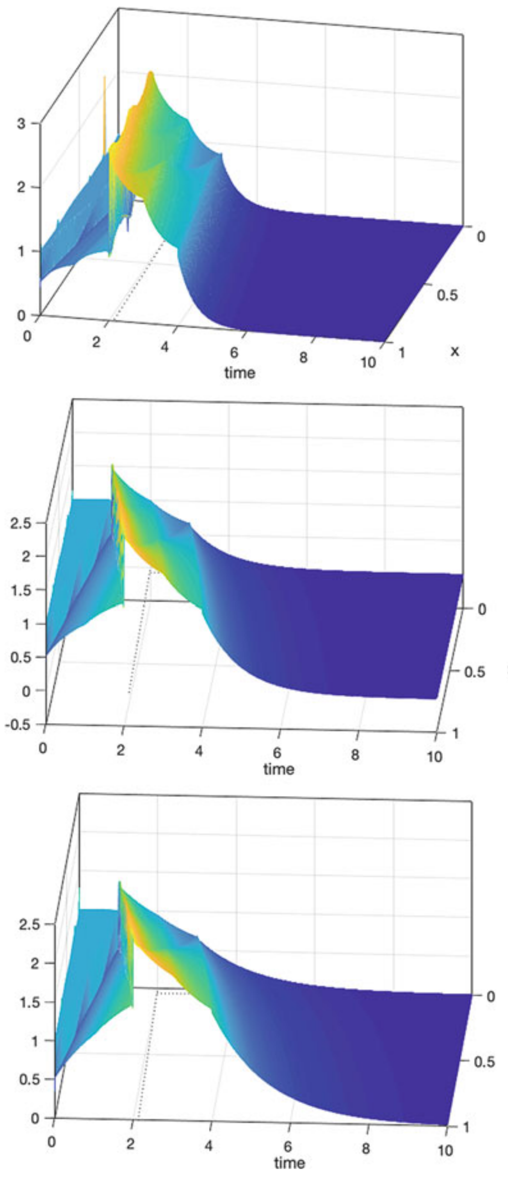


Optimization-Based Control Design Techniques and Tools, Fig. 6 Synthesis at nominal $q_0 = 3$. Simulations of nominal $K = K_0 + \texttt{feedback}(\widetilde{K}, \Phi(3))$ for $q = 2, 3, 4$. Nominal controller is robustly stable over $[\underline{q}, \overline{q}]$

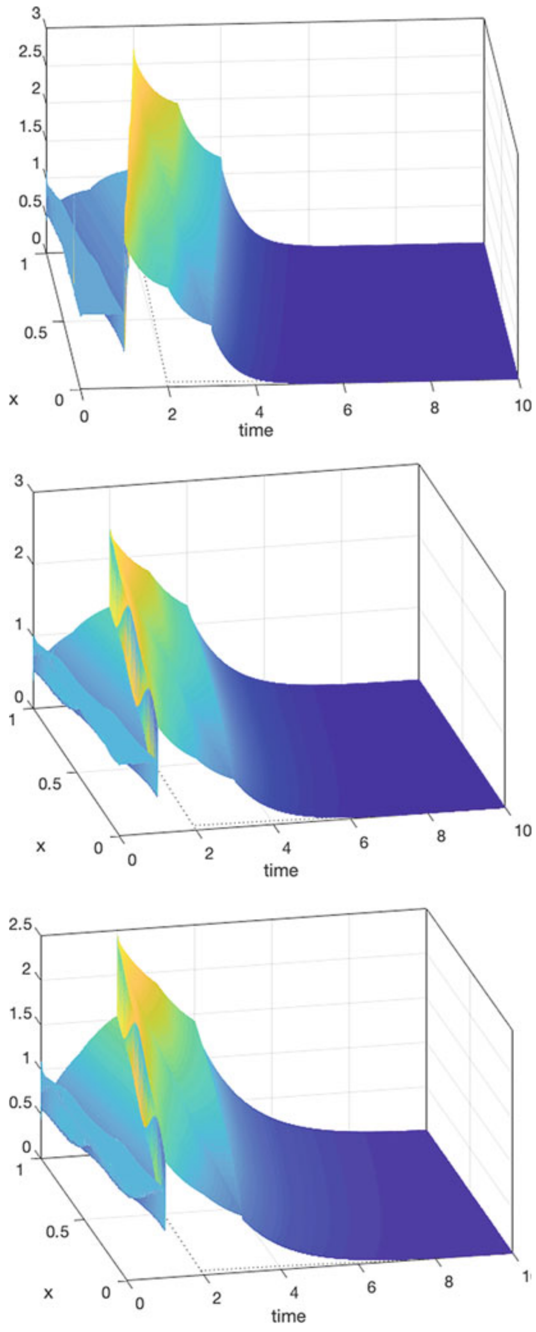
sufficiently slow variations as in Shamma and Athans (1990). We assume that a nominal $q_0 > 0$ and an uncertain interval $[\underline{q}, \overline{q}]$ with $q_0 \in (\underline{q}, \overline{q})$ and $1 \notin [\underline{q}, \overline{q}]$ are given.

The following scheduling scenarios, all leading to implementable controllers, are possible: (a) computing a nominal controller $\widetilde{K}$ at q_0 as before, and scheduling through $\Phi(q)$, which depend explicitly on q, so that $K^{(1)}(q) = K_0 + \texttt{feedback}(\widetilde{K}, -\Phi(q))$, and (b) computing $\widetilde{K}(q)$ which depends on q, and using $K^{(2)}(q) = K_0 + \texttt{feedback}(\widetilde{K}(q), -\Phi(q))$.

While (a) uses (3) based on Apkarian and Noll (2006b, c) and available in SYSTUNE, we show that one can also apply (3) to case (b). We use Fig. 4 to work in the finite-dimensional system $(\widetilde{G}(q), \widetilde{K}(q))$, where plant and controller



Optimization-Based Control Design Techniques and Tools, Fig. 7 Method 1. $\widetilde{K}$ obtained for nominal $q = 3$, but scheduled $K(q) = K_0 + \texttt{feedback}(\widetilde{K}, \Phi(q))$. Simulations for $q = 2$ top, $q = 3$ middle, $q = 4$ bottom



Optimization-Based Control Design Techniques and Tools, Fig. 8 Method 2. $\widetilde{K}(q) = \widetilde{K}_{\text{nom}} + (q - 3)\widetilde{K}_1 + (q - 3)^2\widetilde{K}_2$ and $K(q) = K_0 + \texttt{feedback}(K(q), \Phi(q))$. Simulations for $q = 2, 3, 4$

now depend on q, which is a parameter-varying design.

For that we have to decide on a parametric form of the controller $\widetilde{K}(q)$, which we choose as

$$\widetilde{K}(q,\mathbf{x}) = \widetilde{K}(q_0) + (q-q_0)\widetilde{K}_1(\mathbf{x}) + (q-q_0)^2\widetilde{K}_2(\mathbf{x}),$$

and where we adopted the simple static form $\widetilde{K}_1(\mathbf{x}) = [\mathbf{x}_1\ \mathbf{x}_2\ \mathbf{x}_3]$, $\widetilde{K}_2 = [\mathbf{x}_4\ \mathbf{x}_5\ \mathbf{x}_6]$, featuring a total of 6 tunable parameters. The nominal $\widetilde{K}(q_0)$ is obtained via (3) as above. For $q_0 = 3$ this leads to $\widetilde{K}(q_0) = [-1.049\ \ -1.049\ \ -0.05402]$, computed via SYSTUNE.

With the parametric form $\widetilde{K}(q,\mathbf{x})$ fixed, we now use again the feedback system $(\widetilde{G}(q), \widetilde{K}(q))$ in Fig. 4 and design a parametric robust controller using the method of Apkarian et al. (2015a), which is included in the SYSTUNE package and used by default if an uncertain closed-loop is entered. The tuning goals are chosen as constraints on closed-loop poles including minimum decay of 0.7, minimum damping of 0.9, with maximum frequency 2. The controller obtained is (with $q_0 = 3$)

$$\widetilde{K}(q,\mathbf{x}^*) = \widetilde{K}(q_0) + (q-q_0)\widetilde{K}_1(\mathbf{x}^*) + (q-q_0)^2\widetilde{K}_2(\mathbf{x}^*),$$

with $\widetilde{K}_1 = [-0.1102, -0.1102, -0.1053]$, $\widetilde{K}_2 = [0.03901, 0.03901, 0.02855]$, and we retrieve the final parameter varying controller for $G(q)$ as

$$K^{(2)}(q) = K_0 + \texttt{feedback}(\widetilde{K}(q,\mathbf{x}^*), -\Phi(q)).$$

Nominal and scheduled controllers are compared in simulation in Figs. 6, 7, and 8, which indicate that $K^{(2)}(q)$ achieves the best performance for frozen-in-time values $q \in [2,4]$. All controllers are easily implementable, since only real-rational elements in combination with delays are used.

The non-smooth program (5) was solved with SYSTUNE in 30 s CPU on a Mac OS X with 2.66 GHz Intel Core i7 and 8 GB RAM. The reader is referred to the MATLAB Control Toolbox 2018b and higher versions for additional examples. More details on this study can be found in Apkarian and Noll (2019).

Cross-References

- ▶ H_∞ Control
- ▶ Optimization-Based Robust Control
- ▶ Robust Synthesis and Robustness Analysis Techniques and Tools

Bibliography

Apkarian P (2013) Tuning controllers against multiple design requirements. In: American control conference (ACC), pp 3888–3893

Apkarian P, Noll D (2006a) Controller design via nonsmooth multi-directional search. SIAM J Control Optim 44(6):1923–1949

Apkarian P, Noll D (2006b) Nonsmooth H_∞ synthesis. IEEE Trans Aut Control 51(1):71–86

Apkarian P, Noll D (2006c) Nonsmooth optimization for multidisk H_∞ synthesis. Eur J Control 12(3):229–244

Apkarian P, Noll D (2007) Nonsmooth optimization for multiband frequency domain control design. Automatica 43(4):724–731

Apkarian P, Noll D (2018) Structured H_∞-control of infinite dimensional systems. Int J Robust Nonlinear Control 28(9):3212–3238

Apkarian P, Noll D (2019) Boundary control of partial differential equations using frequency domain optimization techniques. arXiv:1905.06786v1, pp 1–22

Apkarian P, Noll D, Thevenet JB, Tuan HD (2003) A spectral quadratic-SDP method with applications to fixed-order H_2 and H_∞ synthesis. Eur J Control 10(6):527–538

Apkarian P, Noll D, Rondepierre A (2007) Mixed H_2/H_∞ control via nonsmooth optimization. In: Proceedings of the 46th IEEE conference on decision and control, New Orleans, pp 4110–4115

Apkarian P, Noll D, Prot O (2008) A trust region spectral bundle method for nonconvex eigenvalue optimization. SIAM J Optim 19(1):281–306

Apkarian P, Dao MN, Noll D (2015a) Parametric robust structured control design. IEEE Trans Auto Control 60(7):1857–1869

Apkarian P, Noll D, Ravanbod L (2015b) Computing the structured distance to instability. In: SIAM conference on control and its applications, pp 423–430. https://doi.org/10.1137/1.9781611974072.58

Apkarian P, Noll D, Ravanbod L (2016) Nonsmooth bundle trust-region algorithm with applications to robust stability. Set-Valued Var Anal 24(1):115–148

Apkarian P, Noll D, Ravanbod L (2018) Non-smooth optimization for robust control of infinite-dimensional systems. Set-Valued Var Anal 26(2):405–429

Benner P, Sima V, Voigt M (2012) L_∞-norm computation for continuous-time descriptor systems using structured matrix pencils. IEEE Trans Aut Control 57(1):233–238

Boyd S, Balakrishnan V, Kabamba P (1989) A bisection method for computing the H_∞ norm of a transfer matrix and related problems. Math Control Signals Syst 2(3):207–219

Bresch-Pietri D, Krstic M (2014) Output-feedback adaptive control of a wave PDE with boundary anti-damping. Automatica 50(5):1407–1415

Burke J, Lewis A, Overton M (2005) A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM J Optim 15:751–779

DeQueiroz M, Rahn CD (2002) Boundary control of vibrations and noise in distributed parameter systems: an overview. Mech Syst Signal Process 16(1):19–38

Falcoz A, Pittet C, Bennani S, Guignard A, Bayart C, Frapard B (2015) Systematic design methods of robust and structured controllers for satellites. Application to the refinement of Rosetta's orbit controller. CEAS Space J 7:319–334. https://doi.org/10.1007/s12567-015-0099-8

Fares B, Apkarian P, Noll D (2001) An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory. Int J Control 74(4):348–360

Fares B, Noll D, Apkarian P (2002) Robust control via sequential semidefinite programming. SIAM J Control Optim 40(6):1791–1820

Fridman E (2014) Introduction to time-delay systems. Systems and control, foundations and applications. Birkhuser, Basel

Gahinet P, Apkarian P (2011) Structured H_∞ synthesis in MATLAB. In: Proceedings of IFAC world congress, Milan, pp 1435–1440

Ganet-Schoeller M, Desmariaux J, Combier C (2017) Structured control for future european launchers. AeroSpaceLab 13:2–10

Kocvara M, Stingl M (2003) A code for convex nonlinear and semidefinite programming. Optim Methods Softw 18(3):317–333

Kolda TG, Lewis RM, Torczon V (2003) Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev 45(3):385–482

Lemaréchal C (1975) An extension of Davidson's methods to nondifferentiable problems. In: Balinski ML, Wolfe P (eds) Nondifferentiable optimization. Mathematical programming study, Springer, Berlin, Heidelberg, vol 3, pp 95–109

Lemaréchal C, Oustry F (2000) Nonsmooth algorithms to solve semidefinite programs. In: El Ghaoui L, Niculescu S-I (ed) SIAM advances in linear matrix inequality methods in control Series, SIAM, pp 57–77

Lieslehto J (2001) PID controller tuning using evolutionary programming. In: American control conference, vol 4, pp 2828–2833

Moelja AA, Meinsma G (2003) Parametrization of stabilizing controllers for systems with multiple i/o delays. IFAC Proc Vol 36(19):351–356. 4th IFAC workshop on time delay systems (TDS 2003), Rocquencourt, 8–10 Sept 2003

Noll D (2007) Local convergence of an augmented Lagrangian method for matrix inequality constrained programming. Optim Methods Softw 22(5):777–802

Noll D, Apkarian P (2005) Spectral bundle methods for nonconvex maximum eigenvalue functions: first-order methods. Math Prog Ser B 104(2):701–727

Noll D, Torki M, Apkarian P (2004) Partially augmented Lagrangian method for matrix inequality constraints. SIAM J Optim 15(1):161–184

Noll D, Prot O, Rondepierre A (2008) A proximity control algorithm to minimize nonsmooth and nonconvex functions. Pac J Optim 4(3):571–604

Noll D, Prot O, Apkarian P (2009) A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions. J Convex Anal 16(3 & 4):641–666

Oi A, Nakazawa C, Matsui T, Fujiwara H, Matsumoto K, Nishida H, Ando J, Kawaura M (2008) Development of PSO-based PID tuning method. In: International conference on control, automation and systems, pp 1917–1920

Ravanbod L, Noll D, Apkarian P (2017) Branch and bound algorithm with applications to robust stability. J Glob Optim 67(3):553–579

Robust Control Toolbox 42 (2012) The MathWorks Inc. Natick, MA

Saldivar B, Mondié S, Loiseau J, Rasvan V (2013) Suppressing axial-torsional vibrations in drillstrings. J Control Eng Appl Inf, SRAIT 14:3–10

Shamma JF, Athans M (1990) Analysis of gain scheduled control for nonlinear plants. IEEE Trans Aut Control 35(8):898–907

Smyshlyaev A, Krstic M (2009) Boundary control of an anti-stable wave equation with anti-damping on the uncontrolled boundary. Syst Control Lett 58: 617–623

Stein G, Doyle J (1991) Beyond singular values and loopshapes. AIAA J Guid Control 14:5–16

Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River