

# Optimization-Based Control Design Techniques and Tools

Pierre Apkarian, Dominikus Noll

## ▶ To cite this version:

Pierre Apkarian, Dominikus Noll. Optimization-Based Control Design Techniques and Tools. Encyclopedia of Systems and Control, Springer London, pp.1-11, 2015, 10.1007/978-1-4471-5102-9\_144-2. hal-01868381

## HAL Id: hal-01868381 https://hal.science/hal-01868381

Submitted on 5 Sep 2018

**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.

### **Optimization-Based Control Design Techniques and Tools**

Pierre Apkarian

Dominikus Noll

Abstract—Structured output feedback controller synthesis is an exciting new 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  $H_{\infty}$ - or  $H_2$ -control within control architectures preferred by practitioners. The new approach to structured  $H_{\infty}$ -control, developed during the past decade, is rooted in a change of paradigm in the synthesis algorithms. Structured design may no longer be 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.

#### I. MOTIVATIONS

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- 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 dramatic increase in plant complexity is exacerbated by the desire 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. Such practical constraints explain the limited use of blackbox controllers, and they are the driving force for the implementation of *structured* control architectures, as well as for the tendency to replace *hand* tuning methods by rigorous algorithmic optimization tools.

#### **II. STRUCTURED CONTROLLERS**

Before addressing specific optimization techniques, we introduce some basic terminology for control design problems with structured controllers. Given a plant P in state-space form

$$P: \begin{cases} \dot{x} = Ax + B_1w + B_2u \\ z = C_1x + D_{11}w + D_{12}u \\ y = C_2x + D_{21}w + D_{22}u \end{cases}$$
(1)

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, an output feedback controller K in state-space form

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

with  $A_K \in \mathbb{R}^{n_K \times n_K}$ , is called *structured* if the 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  $K = K(\mathbf{x})$ . A structured controller synthesis problem is then an optimization problem of the form

minimize 
$$||T_{wz}(P, K(\mathbf{x}))||$$
  
subject to  $K(\mathbf{x})$  closed-loop stabilizing (3)  
 $K = K(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n$ 

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 [1]. The norm  $\|\cdot\|$  stands for the  $H_{\infty}$ -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:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & -1/\tau & -k_D/\tau \\ \hline k_I & 1/\tau & k_P + k_D/\tau \end{bmatrix}, \begin{bmatrix} A - B_2 K_c - K_f C_2 & K_f \\ \hline -K_c & 0 \end{bmatrix}$$
$$\begin{bmatrix} A_K & B_K \\ \hline C_K & D_K \end{bmatrix}, \begin{bmatrix} \frac{q}{\text{diag}} A_{Ki} & \frac{diag}{q} B_{Ki} \\ \frac{i=1}{q} & \frac{i=1}{q} \\ \frac{diag}{q} C_{Ki} & \frac{diag}{i=1} D_{Ki} \\ \frac{diag}{q} D_{Ki} \end{bmatrix}.$$

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

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

Dominikus Noll is with Université de Toulouse, Institut de Mathmatiques, 118, route de Narbonne, 31062 Toulouse, France dominikus.noll@math.univ-toulouse.fr

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.



Fig. 1. Black-box full-order controller K on the left, structured 2-DOF control architecture with  $K = \text{block-diag}(K_1, K_2)$  on the right.

More sophisticated controllers structures  $K(\mathbf{x})$  arise form architectures like for instance 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  $K(\mathbf{x})$  with

$$P := \begin{bmatrix} A & 0 & 0 & B \\ \hline C & 0 & 0 & D \\ 0 & I & 0 & 0 \\ -C & 0 & I & -D \end{bmatrix}, K(\mathbf{x}) := \begin{bmatrix} K_1(s) & 0 \\ 0 & K_2(s) \end{bmatrix}$$

where  $K(\mathbf{x}, s)$  takes a typical block-diagonal structure featuring the tunable elements  $\mathbf{x} = (a, k_P, k_I)$ .

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



Fig. 2. Synthesis of  $K = \text{block-diag}(K_1, \ldots, K_N)$  against multiple requirements or models  $P^{(1)}, \ldots, P^{(M)}$ . Each  $K_i(\mathbf{x})$  can be structured.

The structure concept is equally useful to deal with the second central challenge in control design: system uncertainty. The latter may be handled with  $\mu$ -synthesis techniques [2] 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 minimize  $f(\mathbf{x}) = \max_{k \in \text{SOFT}, i \in I_k} \|T_{w_i z_i}^{(k)}(K(\mathbf{x}))\|$ subject to  $g(\mathbf{x}) = \max_{k \in \text{HARD}, j \in J_k} \|T_{w_j z_j}^{(k)}(K(\mathbf{x}))\| \le 1 \quad (4)$  $K = K(\mathbf{x}) \text{ closed-loop stabilizing}$  $\mathbf{x} \in \mathbb{R}^n$ 

where  $T_{w_i z_i}^{(k)}$  denotes the *i*th closed-loop robustness or performance channel  $w_i \to z_i$  for the *k*-th plant model  $P^{(k)}(s)$ . The rationale of (4) is to minimize the worst-case cost of the soft constraints  $||T_{w_i z_i}^{(k)}||$ ,  $k \in \text{SOFT}$ , while enforcing the hard constraints  $||T_{w_j z_j}^{(k)}|| \leq 1$ ,  $k \in \text{HARD}$ .

#### **III. OPTIMIZATION TECHNIQUES OVER THE YEARS**

During the late 1990s the necessity to develop design techniques for structured regulators  $K(\mathbf{x})$  was recognized [3], and the limitations of synthesis methods based on algebraic Riccati equations (AREs) or linear matrix inequalities (LMIs) became evident, as these techniques can only provide black-box controllers. Unfortunately, the lack of appropriate synthesis techniques for structured  $K(\mathbf{x})$  led to the unsatisfying situation, where sophisticated approaches like the  $H_{\infty}$  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 fill-in for a rigorous design method which directly computes a reduced-order controller. Similarly, hand-tuning of the parameter 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  $H_\infty$ -synthesis problems (3) were addressed by bilinear matrix inequality (BMI) optimization, which used local optimization techniques based on augmented Lagrangian [4]–[6], sequential semidefinite programming methods [7], [8], and non-smooth methods for BMIs [9], [10]. 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, whose number grows quadratically in  $n_P + n_K$  and represents the bottleneck of this approach.

The epoch-making change occurs with the introduction of non-smooth optimization techniques [11]–[14] 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 [15], derivative-free optimization discussed in [16], [17], particle swarm optimization, see [18] and references therein, and also evolutionary computation techniques [19]. The last three classes do not take advantage of 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, specialized non-smooth techniques are highly efficient in practice, are based on a sophisticated convergence theory, capable of solving medium size problems in a matter of seconds, and are still operational for large size problems with several hundreds of states.

#### IV. 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 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

minimize 
$$f(\mathbf{x})$$
  
subject to  $g(\mathbf{x}) \le 0$  (5)  
 $\mathbf{x} \in \mathbb{R}^n$ 

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 [20]. Bundle methods are used to solve difficult problems in integer programming or in stochastic optimization via Lagrangian relaxation. Extensions of the bundling technique to nonconvex problems like (3) or (4) were first developed in [9], [12]–[14], [21], and in more abstract form, in [22].

Fig. 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 model tradition is assured. At the core of the interaction between inner and outer loop is the management of the proximity control parameter  $\tau$ , 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 stepsize in a linesearch, proximity control allows to do shorter trial steps if agreement of the local model with the true objective function is poor, and to allow 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 a 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 explored to speed up the convergence.



Fig. 3. Flow chart of proximity control bundle algorithm

In the case of the  $H_2$ -norm,  $f(\mathbf{x}) = ||T_{wz}(P, K(\mathbf{x}))||_2$ , computing function value and gradient requires essentially the solution of four Lyapunov equations of size  $n_P + n_K$ , see [23], [24]. For the  $H_{\infty}$ -norm,  $f(\mathbf{x}) = ||T_{wz}(P, K(\mathbf{x}))||_{\infty}$ , function evaluation is based on the Hamiltonian algorithm of [25], [26]. The Hamiltonian matrix is of size  $n_P + n_K$ , 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 [12].

#### V. COMPUTATIONAL TOOLS

The novel non-smooth optimization methods became available to the engineering community since 2010 via the MATLAB Robust Control Toolbox [27], [28]. Routines HINFSTRUCT, LOOPTUNE and SYSTUNE are versatile enough to define and combine tunable blocks  $K_i(\mathbf{x})$ , to build and aggregate design requirements  $T_{wz}^{(k)}$  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 the problem (4) to enhance efficiency, see [13] and [12].

It should be mentioned that design problems with multiple hard constraints are inherently complex. It is well known that even simultaneous stabilization of several plants  $P^{(j)}$  with a structured control law  $K(\mathbf{x})$  is generally NP-complete, so that exhaustive methods are expected to fail even for small to medium problems. The principled decision made in [12], and reflected in the MATLAB routines, is to rely on local optimization techniques instead. This leads to weaker 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 [11]. Key features implemented in the mentioned MATLAB routines are discussed in [13], [28], [29].

#### VI. DESIGN EXAMPLE

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 solving a multi-model, fixed-structure reliable flight control design problem.

In reliable flight control one has to maintain stability and adequate performance not only in nominal operation, but also in various scenarios where the aircraft undergoes outages in elevator and aileron actuators. In particular, wind gusts must be alleviated in all outage scenarios to maintain safety. Variants of this problem are addressed in [30].



Fig. 4. Synthesis interconnection for reliable control

The open loop F16 aircraft in the scheme of Fig. 4 has 6 states, the body velocities u, v, w, pitch, roll, and yaw rates q, p, r. The state is available for control as is the flight-path bank angle rate  $\mu$  (deg/s), the angle of attack  $\alpha$  (deg), and the sideslip angle  $\beta$  (deg). Control inputs are the left and right elevator, left and right aileron, and rudder deflections (deg). The controller consists of two blocks, a  $3 \times 6$  state-feedback gain matrix  $K_x$  in the inner loop, and a  $3 \times 3$  integral gain matrix  $K_i$  in the outer loop, which leads to a total of  $27 = \dim \mathbf{x}$  parameters to tune.

In addition to nominal operation, we consider 8 outage scenarios shown in Table I.

 TABLE I

 Outage scenarios where 0 stands for failure

Outage cases	Diagonal of outage gain				
nominal mode	1	1	1	1	1
right elevator outage	0	1	1	1	1
left elevator outage	1	0	1	1	1
right aileron outage	1	1	0	1	1
left aileron outage	1	1	1	0	1
left elevator and right aileron outage	1	0	0	1	1
right elevator and right aileron outage	0	1	0	1	1
right elevator and left aileron outage	0	1	1	0	1
left elevator and left aileron outage	1	0	1	0	1

The design requirements are as follows:

- Good tracking performance in μ, α, and β with adequate decoupling of the three axes.
- Adequate rejection of wind gusts of 5 m/s.
- Maintain stability and acceptable performance in the face of actuator outage.

Tracking is addressed by an LQG-cost [31], which penal-

izes integrated tracking error e and control effort u via

$$J = \lim_{T \to \infty} E\left(\frac{1}{T} \int_0^T \|W_e e\|^2 + \|W_u u\|^2 dt\right).$$
 (6)

Diagonal weights  $W_e$  and  $W_u$  provide tuning knobs for trade-off between responsiveness, control effort, and balancing of the three channels. We use  $W_e =$ diag(20, 30, 20),  $W_u = I_3$  for normal operation and  $W_e =$ diag(8, 12, 8),  $W_u = I_3$  for outage conditions. Modeldependent weights allow to express the fact that nominal operation prevails over failure cases. Weights for failure cases are used to achieve limited deterioration of performance or of gust alleviation under deflection surface breakdown.

The second requirement, wind gust alleviation, is treated as a hard constraint limiting the variance of the error signal e in response to white noise  $w_g$  driving the Dryden wind gust model. In particular, the variance of e is limited to 0.01 for normal operation and to 0.03 for the outage scenarios.

With the notation of section IV, the functions  $f(\mathbf{x})$ and  $g(\mathbf{x})$  in (5) are  $f(\mathbf{x}) := \max_{k=1,...,9} ||T_{rz}^{(k)}(\mathbf{x})||_2$  and  $g(\mathbf{x}) := \max_{k=1,...,9} ||T_{w_ge}^{(k)}(\mathbf{x})||_2$ , where r denotes the setpoint inputs in  $\mu$ ,  $\alpha$  and  $\beta$ . The regulated output z is

$$\boldsymbol{z}^T := \left[ \begin{array}{cc} (\boldsymbol{W}_e^{1/2} \boldsymbol{e})^T & (\boldsymbol{W}_u^{1/2} \boldsymbol{u})^T \end{array} \right]^T \, ,$$

with  $\mathbf{x} = (\operatorname{vec}(K_i), \operatorname{vec}(K_x)) \in \mathbb{R}^{27}$ . Soft constraints are the square roots of J in (6) with appropriate weightings  $W_e$  and  $W_u$ , hard constraints the RMS values of e, suitably weighted to reflect variance bounds of 0.01 and 0.03. These requirements are covered by the Variance and WeightedVariance options in [27].

With this setup, we tuned the controller gains  $K_i$  and  $K_x$ for the nominal scenario only (nominal design) and for all 9 scenarios (fault-tolerant design). The responses to setpoint changes in  $\mu$ ,  $\alpha$ , and  $\beta$  with a gust speed of 5 m/s are shown in Fig. 5 for the nominal design and in Fig. 6 for the faulttolerant design. As expected, nominal responses are good but notably deteriorate when faced with outages. In contrast, the fault-tolerant controller maintains acceptable performance in outage situations. Optimal performance (square root of LQG cost J in (6)) for the fault-tolerant design is only slightly worse than for the nominal design (26 vs. 23). The nonsmooth program (5) was solved with SYSTUNE and the faulttolerant design (9 models, 11 states, 27 parameters) took 30 seconds on Mac OS X with 2.66 GHz Intel Core i7 and 8 GB RAM. The reader is referred to [27] or higher versions, further examples, and additional details.



Fig. 5. Responses to step changes in  $\mu$ ,  $\alpha$  and  $\beta$  for nominal design.



Fig. 6. Responses to step changes in  $\mu$ ,  $\alpha$  and  $\beta$  for fault-tolerant design.

#### **CROSS REFERENCES**

#### References

- A. Varga and G. Looye, "Symbolic and numerical software tools for lft-based low order uncertainty modeling," in *In Proc. CACSD'99 Symposium, Cohala*, 1999, pp. 1–6.
- [2] G. Stein and J. Doyle, "Beyond singular values and loopshapes," AIAA Journal of Guidance and Control, vol. 14, pp. 5–16, 1991.
- [3] B. Fares, P. Apkarian, and D. Noll, "An Augmented Lagrangian Method for a Class of LMI-Constrained Problems in Robust Control Theory," *Int. J. Control*, vol. 74, no. 4, pp. 348–360, 2001.
- [4] —, "An Augmented Lagrangian Method for a Class of LMI-Constrained Problems in Robust Control Theory," in *Proc. American Control Conf.*, Chicago, Illinois, 2000, pp. 3702–3705.
- [5] D. Noll, M. Torki, and P. Apkarian, "Partially augmented lagrangian method for matrix inequality constraints," *submitted*, 2002, rapport Interne, MIP, UMR 5640, Maths. Dept. - Paul Sabatier University.
- [6] M. Kocvara and M. Stingl, "A code for convex nonlinear and semidefinite programming," *Optimization Methods and Software*, vol. 18, no. 3, pp. 317–333, 2003.
- [7] B. Fares, D. Noll, and P. Apkarian, "Robust Control via Sequential Semidefinite Programming," *SIAM J. on Control and Optimization*, vol. 40, no. 6, pp. 1791–1820, 2002.
- [8] P. Apkarian, D. Noll, J. B. Thevenet, and H. D. Tuan, "A Spectral Quadratic-SDP Method with Applications to Fixed-Order  $H_2$  and  $H_{\infty}$  Synthesis," *European J. of Control*, vol. 10, no. 6, pp. 527–538, 2003.
- [9] D. Noll, O. Prot, and P. Apkarian, "A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions," *Journal of Convex Analysis*, vol. 16, no. 3 & 4, pp. 641–666, 2009.
- [10] C. Lemaréchal and F. Oustry, "Nonsmooth algorithms to solve semidefinite programs," SIAM Advances in Linear Matrix Inequality Methods in Control series, ed. L. El Ghaoui & S.-I. Niculescu, 2000.

- [11] D. Noll and P. Apkarian, "Spectral bundle methods for nonconvex maximum eigenvalue functions: first-order methods," *Mathematical Programming Series B*, vol. 104, no. 2, pp. 701–727, Nov. 2005.
- [12] P. Apkarian and D. Noll, "Nonsmooth  $H_{\infty}$  synthesis," *IEEE Trans.* Aut. Control, vol. 51, no. 1, pp. 71–86, 2006.
- [13] —, "Nonsmooth optimization for multiband frequency domain control design," *Automatica*, vol. 43, no. 4, pp. 724–731, April 2007.
- [14] —, "Nonsmooth optimization for multidisk  $H_{\infty}$  synthesis," *European J. of Control*, vol. 12, no. 3, pp. 229–244, 2006.
- [15] J. V. Burke, D. Henrion, A. S. Lewis, and M. L. Overton, "Stabilization via nonsmooth, nonconvex optimization," *IEEE Trans. Aut. Control*, vol. 51, no. 11, pp. 1760–1769, Nov. 2006.
- [16] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: new perspectives on some classical and modern methods," *SIAM Review*, vol. 45, no. 3, pp. 385–482, 2003.
- [17] P. Apkarian and D. Noll, "Controller design via nonsmooth multidirectional search," SIAM J. on Control and Optimization, vol. 44, no. 6, pp. 1923–1949, 2006.
- [18] A. Oi, C. Nakazawa, T. Matsui, H. Fujiwara, K. Matsumoto, H. Nishida, J. Ando, and M. Kawaura, "Development of PSObased PID tuning method," in *International Conference on Control, Automation and Systems*, oct. 2008, pp. 1917 –1920.
- [19] J. Lieslehto, "PID controller tuning using evolutionary programming," in American Control Conference, vol. 4, 2001, pp. 2828 –2833 vol.4.
- [20] C. Lemarechal, "An extension of Davidon methods to nondifferentiable problems," in *Math. Programming Study 3, Nondifferentiable Optimization*, M. L. Balinski and P. Wolfe, Eds. North-Holland, 1975, pp. 95–109.
- [21] P. Apkarian, D. Noll, and O. Prot, "A trust region spectral bundle method for nonconvex eigenvalue optimization," *SIAM J. on Optimization*, vol. 19, no. 1, pp. 281–306, 2008.
- [22] D. Noll, O. Prot, and A. Rondepierre, "A proximity control algorithm to minimize nonsmooth and nonconvex functions," *Pacific J. of Optimization*, vol. 4, no. 3, pp. 571–604, 2008.
- [23] P. Apkarian, D. Noll, and A. Rondepierre, "Mixed  $H_2/H_{\infty}$  control via nonsmooth optimization," in *Proc. of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, 2007, pp. 4110–4115.
- [24] T. Rautert and E. W. Sachs, "Computational design of optimal output feedback controllers," *SIAM Journal on Optimization*, vol. 7, no. 3, pp. 837–852, 1997.
- [25] P. Benner, V. Sima, and M. Voigt, " $L_{\infty}$ -norm computation for continuous-time descriptor systems using structured matrix pencils," *IEEE Trans. Aut. Control*, vol. 57, no. 1, pp. 233–238, 2012.
- [26] S. Boyd, V. Balakrishnan, and P. Kabamba, "A bisection method for computing the  $H_{\infty}$  norm of a transfer matrix and related problems," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 3, pp. 207–219, 1989.
- [27] Robust Control Toolbox 4.2, "The MathWorks Inc. Natick, MA, USA," 2012.
- [28] P. Gahinet and P. Apkarian, "Structured  $H_{\infty}$  synthesis in MATLAB," in *Proc. IFAC World Congress*, Milan, Italy, 2011, pp. 1435–1440.
- [29] P. Apkarian, "Tuning controllers against multiple design requirements," Proc. American Control Conf., 2013.
- [30] F. Liao, J. L. Wang, and G.-H. Yang, "Reliable robust flight tracking control: An LMI approach," *IEEE Trans. Control Sys. Tech.*, vol. 10, pp. 76–89, 2002.
- [31] J. M. Maciejowski, *Multivariable Feedback Design*. Addison-Wesley, 1989.