Modulating robustness in robust control: making it easy through randomization

Marco C. Campi Simone Garatti

***Abstract*— Reportedly, robust control can lead to designs that are overconservative because all emphasis is placed on safe-guarding the designed closed-loop against all possible doomy occurrences, and this is done at the price of sacrificing performance. When 100% guarantee of robustness is required, standard robust control is indeed the way to go. However, in many applications, robustness in 100% of the cases is not really necessary and it is a fact that accepting a small compromise in robustness guarantees (e.g. accepting a 99% guarantee) can often times lead to a huge improvement in performance.**
At the present stage of knowledge, the real stumbling-block is the lack of computationally-tractable algorithmic methods to work out 99%-guaranteed solutions trading the remaining 1% of guarantee for performance. This paper aims at opening new directions to solve this problem, and we show that this result can be achieved through randomization.

I. Introduction

Many robust control problems can be cast as optimization programs where the figure of merit expresses performance and the constraints (usually parameterized by the uncertain plant parameters) represent the robustness requirements. In many – albeit not all – cases, the program is convex (e.g. all problems with LMI constraints), see [4], [5], [10]. This note refers to this latter class of problems and the approach presented herein applies under the assumption of convexity.

A. Robust programs

In general terms, a robust program for controller design is as follows

$$\begin{aligned} \text{RP}: \quad & \min_{\alpha\in\mathbb{R}^d} c^T\alpha \\ & \text{subject to: } f_\delta(\alpha)\le 0, \quad \forall\delta\in\Delta, \end{aligned} \tag{1}$$

where α is the vector of the controller parameters, and $f_\delta(\alpha)\le 0$ is a family of convex constraints parameterized by δ, the vector of uncertain parameters of the plant. Requiring that $f_\delta(\alpha)\le 0, \forall\delta\in\Delta$, delivers full guarantee that – whatever the plant is – the solution does not violate the corresponding requirement. Note that linearity of the objective function is without loss of generality since any problem of the kind $\min_{\alpha\in\mathbb{R}^d} c(\alpha)$, where $c(\alpha)$ is a convex function, can be re-written as (here q is a slack variable) $\min_{q\in\mathbb{R},\alpha\in\mathbb{R}^d} q$, provided that constraint $c(\alpha)\le q$ is added to the other constraints.

Paper supported by the MIUR national project "Identification and adaptive control of industrial systems"

M.C. Campi is with the Dipartimento di Elettronica per l'Automazione, Università di Brescia, via Branze 38, 25123 Brescia, Italia . E-mail: campi@ing.unibs.it

S. Garatti is with the Dipartimento di Elettronica ed Informazione, Politecnico di Milano, p.zza L. da Vinci 32, 20133 Milano, Italy. E-mail: sgaratti@elet.polimi.it

*Example 1 (*feed-forward pole/zero cancellation*): Consider the discrete-time problem depicted in Figure 1, where one is required to design a controller* $1+\alpha z^{-1}$ *so*

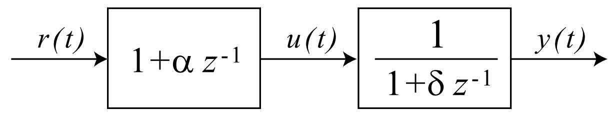

Fig. 1. A feed-forward compensation problem.

as to achieve a transfer function $\frac{1+\alpha z^{-1}}{1+\delta z^{-1}}$ *from* $r(t)$ *to* $y(t)$ *as close as possible to the identity transfer function in the* 2*-norm sense.*
The plant is uncertain in that the plant pole δ *is only known to belong to a certain set of feasibility:* $\delta\in\Delta\subseteq\mathbb{R}$*. Then, the goal is to find a controller which minimizes the* 2*-norm in the worst case. In mathematical terms, this amounts to solve the following robust program:*

$$\min_{\alpha\in\mathbb{R}}\max_{\delta\in\Delta}\left\|\frac{1+\alpha z^{-1}}{1+\delta z^{-1}}-1\right\|_2^2 =$$
$$\min_{\alpha\in\mathbb{R}}\max_{\delta\in\Delta}\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\frac{1+\alpha e^{-j\omega}}{1+\delta e^{-j\omega}}-1\right|^2 d\omega,$$

which, in turn, can be rewritten as an RP *of the form* (1)*:*

$$\begin{aligned} & \min_{q\in\mathbb{R},\alpha\in\mathbb{R}} q \\ & \text{subject to: } \left\|\frac{1+\alpha z^{-1}}{1+\delta z^{-1}}-1\right\|_2^2 \le q, \quad \forall\delta\in\Delta. \end{aligned} \tag{2}$$

To see this, note that, given an α*, the slack variable* q *represents an upper bound on the min-max problem cost* $\left\|\frac{1+\alpha z^{-1}}{1+\delta z^{-1}}-1\right\|_2^2$ *achieved when* δ *ranges over the uncertainty set* Δ*. By solving* (2) *we seek that* α *which corresponds to the smallest upper bound* q*.*

One recognized drawback of robust convex programs is that they tend to return *conservative* solutions, because the solution is determined by a few "ill" plant instances in the uncertain domain, [2], [13], [14], [16]. For instance, in the setting of Example 1, if $\Delta=\{-0.9\}\cup[0.85,0.95]$ (i.e. all possible plants have a positive pole located around 0.9 with the sole exception of a single case where the pole is -0.9), no high-performing compensator can be found according to (2), although it is apparent that the controller $1+0.9z^{-1}$ attains a good result in all cases except just one.

B. Randomized approach

Dealing with an infinite number of constraints (one constraint for each $\delta \in \Delta$ where Δ is usually a continuous domain containing an infinite number of possible δ occurrences) is hard, [3], [6]. For this reason, a stream of literature has recently sprung up where the infinite wealth of occurrences in Δ is tamed by only concentrating on a finite number of possible situations, picked at random from the original set Δ, see [8]. The goal of these contributions is to get as close as possible to the robust solution by following a randomized route which is alternative to standard deterministic methods and which can be pursued at a doable computational effort. This paper is cast in this same line of research, but opens up new horizons in the direction of enlightening procedures for achieving a different objective: that of trading robustness for performance.

Suppose that N independent and identically distributed plant parameter vectors $\delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(N)}$ are extracted from Δ according to some user-chosen probability Pr (sampling or randomization of Δ). Depending on the situation at hand, Pr can have different interpretations. Sometimes, it is the actual probability with which uncertainty parameters occur. Other times, Pr simply describes the relative importance we attribute to different uncertainty plant instances. In the randomized approach of [8], it is suggested to concentrate on the extracted $\delta^{(i)}$'s only and to perform optimization with only the corresponding constraints in place:

$$\begin{aligned} \mathrm{RP}_N: \quad & \min_{\alpha \in \mathbb{R}^d} c^T \alpha \\ & \text{subject to: } f_{\delta^{(i)}}(\alpha) \leq 0, \quad i = 1, \ldots, N. \end{aligned}$$

This seemingly naive approach finds a solid mathematical reason of being in a theory that provides guarantees about the level of violation of the so constructed solution with respect to all other unseen constraints in the Δ set (see [8], Theorem 1).

In this paper we take a fundamental step forward with respect to the approach of [8]. Along this latter, indeed, after extracting $\delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(N)}$, one is facing with a finite optimization problem where all N constraints have to be satisfied; again, as in the original robust problem RP, just few constraints usually determine the solution and the solution is on the conservative side. Breaking up with this paradigm, we here allow the user to a-posteriori discard some of the constraints, those that are more adverse to the control objective. This way, the optimization cost is improved, at times by a large quantity. The fundamental fact established in this paper is that the so-found solution is still robust to an extent that can be modulated by the number of constraints that are actually discarded, in other words the portion of constraints in the original Δ set that are possibly violated by the solution is kept under control by a new theory as introduced in this paper.

One additional remark of practical importance is that the procedure according to which constraints are discarded is totally arbitrary as far as the theoretical result about violation is concerned. This is important because, normally, discarding the constraints that at best improve the control objective is not easy and it is in fact a hard combinatorial problem. Here, the user is not required to discard the constraints optimally, he/she is allowed to use his/her favorite algorithm, perhaps a greedy one, and still the theory remains valid.

II. VIOLATION RESULTS

Let $k < N$ and let $\mathscr{A}$ be a *constraints removal algorithm* through which k constraints are discarded from $\delta^{(1)}, \ldots, \delta^{(N)}$. The output of $\mathscr{A}$ is the set $\mathscr{A}(\delta^{(1)}, \ldots, \delta^{(N)}) = \{i_1, \ldots, i_{N-k}\}$ of $N-k$ indexes from $\{1, \ldots, N\}$ representing the constraints still in place. Consider then the following randomized program where k constraints are removed as indicated by $\mathscr{A}$:

$$\begin{aligned} \mathrm{RP}_{N,k}^{\mathscr{A}}: \quad & \min_{\alpha \in \mathbb{R}^d} c^T \alpha \\ & \text{subject to: } f_{\delta^{(i)}}(\alpha) \leq 0, \; i \in \mathscr{A}(\delta^{(1)}, \ldots, \delta^{(N)}). \end{aligned}$$

$\mathscr{A}$ can e.g. be a greedy removal algorithm where one selects in succession those constraints which – if removed one by one – lead each time to the largest immediate improvement in the objective function. This greedy approach has the great advantage of being implementable at a low computational effort.
Once the solution of $\mathrm{RP}_{N,k}^{\mathscr{A}}$, say $\alpha_{N,k}$, has been found, one can inspect the incurred control objective value $c^T \alpha_{N,k}$ for satisfaction, while the following Theorem 1 provides theoretical guarantees that the solution violates less than a fraction γ of the total amount of constraints in Δ with probability $1-\beta$.

Theorem 1: Fix two real numbers $\gamma \in (0,1)$ (violation level) and $\beta \in (0,1)$ (confidence level). If N and k are such that $N-k > d$ and

$$\binom{N}{d} \sum_{i=0}^{k} \binom{N-d}{i} \gamma^i (1-\gamma)^{N-d-i} \leq \beta, \tag{3}$$

then, independently of the removal algorithm $\mathscr{A}$, with probability at least $1-\beta$, we have that

$$\Pr\{\delta \in \Delta : f_\delta(\alpha_{N,k}) > 0\} \leq \gamma. \tag{4}$$

Proof: *see [9]. The proof is also available on request from the authors.*

Remark 1: In simple words, equation (4) says that the designed controller corresponding to $\alpha_{N,k}$ is robust up to level γ, that is the robustness requirements are violated for at most a γ-fraction of the plants.
As for probability $1-\beta$, one should note that $\alpha_{N,k}$ is a random element depending on the extracted $\delta^{(1)}, \ldots, \delta^{(N)}$. Therefore, the violation probability $\Pr\{\delta \in \Delta : f_\delta(\alpha_{N,k}) > 0\}$ is a random variable too, and it can satisfy (4) for some constraints extractions and not for others. β refers to the probability of observing a "bad" multi-sample $\delta^{(1)}, \ldots, \delta^{(N)}$

such that (4) does not hold.

Remark 2: Formula (3) can be used to design an experiment, where one wishes to a-priori fix desired levels for γ and β, as well as the number k of constraints one wants to remove to a-posteriori improve the control cost, and then determine the number N of constraints necessary for achieving these levels of violation and confidence. To this end, N satisfying (3) is determined via numerical computation. For the purpose of illustration, some values of N returned by (3) with $k=10$ and $d=2$ for some typical values of β and γ are given in Table I.

β vs. γ	0.1	0.05	0.01
10^{-3}	387	828	4647
10^{-6}	476	1008	5541
10^{-9}	560	1179	6398

TABLE I

N GIVEN BY (3) WHEN $k=10$ AND $d=2$.

It is perhaps worth noticing that the dependence of N on β is logarithmic, so that small values of β (10^{-10} or even 10^{-20}) can be forced in without affecting too much the number of constraints to be extracted. This means that, for practical purposes, β has a very marginal relevance.

A. Comparison with chance-constrained optimization

Stronger results can be established when algorithm $\mathscr{A}$ is optimal (call it $\mathscr{A}^*$), that is, among the extracted $\delta^{(1)},\dots,\delta^{(N)}$, the k constraints generating the largest improvement in the objective function are left out. In this latter case, $\mathrm{RP}^{\mathscr{A}}_{N,k}$ becomes the following program:

$$\mathrm{RP}^{\mathscr{A}^*}_{N,k}: \quad \min_{\alpha\in\mathbb{R}^d} c^T\alpha \quad \text{subject to:} \quad f_{\delta^{(i)}}(\alpha)\leq 0 \text{ for at least } N-k \text{ constraints out of the } N \text{ extracted ones,} \tag{5}$$

where optimization is intended to be performed not only over α but also over the k constraints to be discarded. That is, one is required to discard the constraints whose removal achieves the largest improvement in performance as compared to any other possible removal of k constraints.

$\mathrm{RP}^{\mathscr{A}^*}_{N,k}$ is the counterpart with *finite* constraints of the so-called chance-constrained optimization program CCP_γ associated to the RP in (1), see [12]:

$$\mathrm{CCP}_\gamma: \quad \min_{\alpha\in\mathbb{R}^d} c^T\alpha \quad \text{subject to: } f_\delta(\alpha)\leq 0 \text{ with probability } \Pr\geq 1-\gamma. \tag{6}$$

Thus, CCP_γ is an optimization program with infinite constraints where however constraint violation is tolerated in order to alleviate the conservatism of RP. The portion of violated constraints, however, must be no larger than γ, and it has to be optimally chosen so as to achieve the best improvement in the objective function.

Letting α^*_γ be the optimal solution of CCP_γ, the optimal objective value $c^T\alpha^*_\gamma$ is a decreasing function of γ and provides a quantification of the trade-off between robustness and performance.

Theorem 1 establishes a fundamental link between the chance-constrained program (6) and the randomized program (5). Indeed, Theorem 1 says that the solution $\alpha^*_{N,k}$ of $\mathrm{RP}^{\mathscr{A}^*}_{N,k}$ is feasible for CCP_γ with large probability $1-\beta$, provided that (3) holds. The following theorem further links the two problems by relating their optimal cost.

Theorem 2: Fix two real numbers $\varepsilon\in(0,1)$ (accuracy level) and $\beta'\in(0,1)$ (confidence level). If N and k are such that $N-k>d$ and

$$\sum_{i=k+1}^{N}\binom{N}{i}(\gamma-\varepsilon)^i(1-\gamma+\varepsilon)^{N-i}\leq\beta', \tag{7}$$

then, with probability at least $1-\beta'$, we have that

$$c^T\alpha^*_{N,k}\leq c^T\alpha^*_{\gamma-\varepsilon}.$$

Proof: *see [9]. The proof is also available on request from the authors.*

Simply put, Theorem 2 says that the control cost achieved by solving $\mathrm{RP}^{\mathscr{A}^*}_{N,k}$ is no worse than the performance of $\mathrm{CCP}_{\gamma-\varepsilon}$, where ε is a degradation margin.

Remark 3: The optimal removal of k constraints among the N initial ones is a nontrivial combinatorial problem. Indeed, a brute-force approach (where one solves the optimization problems for all possible combinations of $N-k$ constraints taken out from the initial set of N constraints and then choose that combination resulting in the lowest value of the objective function) requires to solve $\binom{N}{k}$ optimization problems, a truly large number in general.

In [11], [1] the point is made that constraints can be removed sequentially by choosing each time among the so-called support constraints only. A support constraint is a constraint whose elimination improves the objective function (see Appendix A for a formal definition). It has been proven in [7], Theorem 2, that among a finite set of constraints at most d can be of support (d, we recall, is the dimensionality of the optimization variable α). This observations reduces the actual number of possible combinations of constraints to be taken into account in the problem of optimally discarding k constraints, resulting in an algorithm that require to solve $O(\min\{N\cdot d^k, N\cdot k^d\})$ optimization problems only.

Admittedly, however, even for relatively small values of d (say e.g. $d=10$), $N\cdot k^d$ and $N\cdot d^k$ grow rapidly with k, and the algorithm complexity becomes intractable even for simple optimization problems (e.g. linear or quadratic programs) for which efficient solvers are available, [5], [15]. This sets practical limits to the use of the optimal algorithm $\mathscr{A}^*$.

III. SIMULATION RESULTS

We conclude the paper with a simple 1-dimensional example which helps gain insight in the presented results.

Let us consider the problem:

$$\min_{\alpha \in \mathbb{R}} \alpha$$
$$\text{subject to: } \alpha \geq \delta, \quad \delta \in [0,1].$$

Here, $f_\delta(\alpha) = \delta - \alpha$ with $\delta \in \Delta = [0,1]$. Suppose also that Pr is uniform over Δ.

For brevity, we henceforth let $V(\alpha) = \Pr\{\delta \in \Delta : f_\delta(\alpha) > 0\}$.

The CCP$_\gamma$ optimum is achieved by removing the set $[1-\gamma, 1]$ from Δ, leading to the optimal solution $\alpha^*_\gamma = 1-\gamma$ and to $V(\alpha^*_\gamma) = \gamma$.

In this 1-dimensional setting, the greedy algorithm for constraints removal is also the optimal algorithm $\mathscr{A}^*$ and given a multi-extraction $\delta^{(1)}, \ldots, \delta^{(N)}$, the optimal solution is obtained by removing the k largest $\delta^{(i)}$'s and by letting $\alpha^*_{N,k}$ to be equal to the $(k+1)^{\text{th}}$ $\delta^{(i)}$ value. Also, $V(\alpha^*_{N,k}) = 1 - \alpha^*_{N,k}$ and the optimal cost is $\alpha^*_{N,k}$.

First, consider the case $k = 0$, i.e no constraints are removed. Figure 2 depicts the probability distribution of $V(\alpha^*_{N,0})$, where we have preferred to display probability in the x-axis, instead of in the y-axis as it is more often done. In this way, given an interval over the x-axis, its length

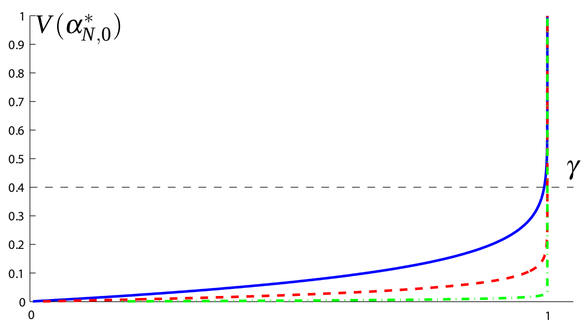

Fig. 2. $V(\alpha^*_{N,0})$ against probability for $N = 10$ (solid line), $N = 30$ (dashed line), $N = 200$ (dashed-dotted line).

represents probability over the multi-extractions domain Δ^N and the vertical values represent the corresponding $V(\alpha^*_{N,0})$ values.

Fix any γ, $\gamma = 0.4$ for example. As it appears, the portion of multi-extractions for which $V(\alpha^*_{N,0}) > \gamma$ rapidly becomes smaller and smaller as N increases, i.e. feasibility for CCP$_\gamma$ is attained with probability rapidly approaching 1.

On the other hand, however, when $V(\alpha^*_{N,0}) \leq \gamma$, $V(\alpha^*_{N,0})$ is much lower than γ for most multi-extractions. This means that the violation of $\alpha^*_{N,0}$ will be much less than that for α^*_γ with high probability, entailing that the objective value of $\alpha^*_{N,0}$ will be poor as compared to the chance-constrained solution, where violation γ is allowed for. From this, we see that $k = 0$ (no constraint removal) is unsuitable, and discarding constraints is necessary for simultaneously securing constraint satisfaction and performance.

Consider now a fixed N, say $N = 200$, and let us have a look at what happens for $k \neq 0$. As Figure 3 shows,

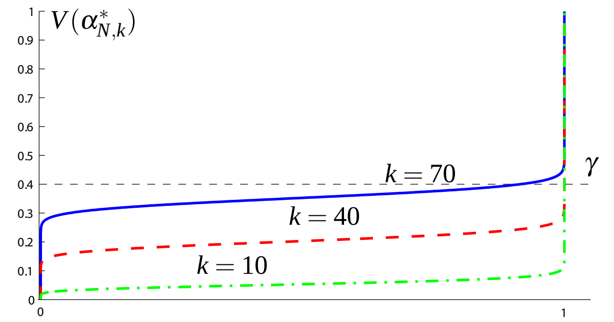

Fig. 3. $V(\alpha^*_{N,k})$ for $N = 200$ and $k = 10, 40, 70$.

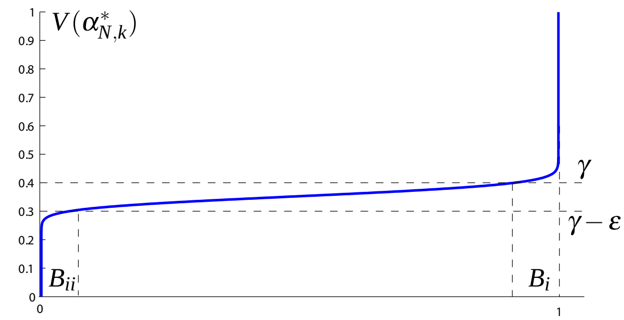

Fig. 4. $V(\alpha^*_{N,k})$ for $N = 200$ and $k = 70$.

while k is let to increase the curve representing $V(\alpha^*_{N,k})$ gets flat at a vertical value that increases with k. For $k = 70$, most multi-extractions have a violation probability near $\gamma = 0.4$. Precisely, violation between $\gamma - \varepsilon = 0.3$ and $\gamma = 0.4$ is achieved with probability 0.83 (see Figure 4 where a violation outside $[0.3, 0.4]$ occurs in $B_i \cup B_{ii}$). Correspondingly, $c^T \alpha^*_{N,k} = 1 - V(\alpha^*_{N,k}) \leq 0.7 = c^T \alpha^*_{\gamma-\varepsilon}$ with high probability (inspect Figure 4 again where $c^T \alpha^*_{N,k} \geq 0.7$ in B_{ii} only). This flatness behavior is at the basis of the fact that feasibility and performance can be simultaneously achieved by discarding constraints.

Thus, we have seen by direct inspection that $N = 200$ and $k = 70$ suffice to simultaneously guarantee

$$\begin{array}{c} V(\alpha^*_{N,k}) \leq 0.4 \\ c^T \alpha^*_{N,k} \leq c^T \alpha^*_{0.3}, \end{array} \qquad (8)$$

with probability 0.83.
By applying Theorem 1 and Theorem 2 with $d = 1$, $\gamma = 0.4$, $\varepsilon = 0.1$, we find that $N = 592$ and $k = 196$ lead to $\beta = 0.13$ and $\beta' = 0.04$ so that result in (8) is guaranteed to hold with probability $1 - \beta - \beta' = 0.83$ as before. While there is a gap between the actual N and k and those given by Theorem 1 and Theorem 2, the strength of these theorems is that they are valid for every convex optimization problem.

REFERENCES

[1] E. Bai, H. Cho, and R. Tempo. Optimization with few violated constraints for linear bounded error parameter estimation. *IEEE Transaction on Automatic Control*, 47:1067–1077, 2002.

[2] B.R. Barmish and C.M. Lagoa. The uniform distribution: A rigorous justification for its use in robustness analysis. *Mathematics of Control, Signals, and Systems*, 10:203–222, 1997.

[3] V.D. Blondel and J.N. Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36:1249–1274, 2000.

[4] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[5] S. Boyd and V. Vandenberghe. *Convex Optimization*. Cambridge University press, Cambridge, UK, 2004.

[6] R.P. Braatz, P.M. Young, J.C. Doyle, and M. Morari. Computational complexity of μ calculation. *IEEE Transactions on Automatic Control*, AC-39:1000–1002, 1994.

[7] G. Calafiore and M.C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[8] G. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[9] M.C. Campi and S. Garatti. Quantitative links between semi-infite chance-constrained optimization and finite chance-constrained optimization. *Submitted for publication*, 2007.

[10] P. Gahinet. Explicit controller formulas for LMI-based H_∞ synthesis. *Automatica*, 32(7):1007–1014, 1996.

[11] J. Matoušek. On geometric optimization with few violated constraints. *Discrete & Computational Geometry*, 14:365–384, 1995.

[12] A. Prèkopa. *Stochastic programming*. Kluwer, Boston, MT, USA, 1995.

[13] R.F. Stengel and L.R. Ray. Stochastic robustness of linear time-invariant control systems. *IEEE Transactions on Automatic Control*, 36:82–87, 1991.

[14] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized algorithms for analysis and control of uncertain systems*. Springer-Verlag, London, UK, 2005.

[15] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[16] M. Vidyasagar. Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica*, 37(10):1515–1528, 2001.

APPENDIX

A. Support constraints

Consider a generic optimization problem with a finite number of constraints:

$$\begin{aligned} \mathrm{P}: \quad & \min_{\alpha\in\mathbb{R}^d} c^T\alpha \\ & \text{subject to: } f_i(\alpha)\le 0, \quad i=1,\dots,N. \end{aligned}$$

Let $\overline{\alpha}$ be the optimal solution (existence and uniqueness of $\overline{\alpha}$, as well as of the solution of all other problems, are assumed here for granted).

Definition 1 (support constraint): The $r-th$ constraint $f_r(x)\le 0$ is a *support constraint* for P if $c^T\overline{\alpha}_r < c^T\overline{\alpha}$, where $\overline{\alpha}_r$ is the optimal solution of the program $\mathrm{P_r}$ obtained from P by removing the $r-th$ constraint, namely:

$$\begin{aligned} \mathrm{P}_r: \quad & \min_{\alpha\in\mathbb{R}^d} c^T\alpha \\ & \text{subject to: } f_i(\alpha)\le 0, \quad i=1,\dots,r-1,r+1,\dots,N. \end{aligned}$$

□

The following theorem has been proved in [7], Theorem 2.

Theorem 3: If $f_i(\alpha)$, $i=1,\dots,N$, are convex functions, then the number of support constraints for P_r is at most d, the size of α.