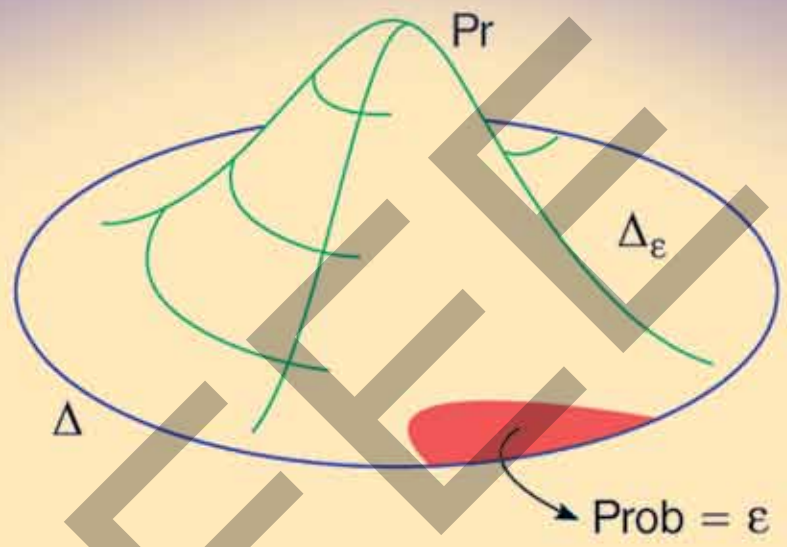
Modulating Robustness in Control Design



PRINCIPLES AND ALGORITHMS

SIMONE GARATTI and MARCO C. CAMPI

Many problems in systems and control, such as controller synthesis and state estimation, are often formulated as optimization problems. In many cases, the cost function incorporates variables that are used to model uncertainty, in addition to optimization variables, and this article employs uncertainty described as probabilistic variables. In a probabilistic setup, a cost value can only be guaranteed with a certain probability. Like pulling down one end of a rope wrapped around a pulley lifts the other end, decreasing the probability improves the cost value. This article analyzes this trade-off and describes quantitative tools to drive the user's choice toward a suitable compromise.

Digital Object Identifier 10.1109/MCS.2012.2234964
Date of publication:

1066-033X/13/$31.00©2013IEEE

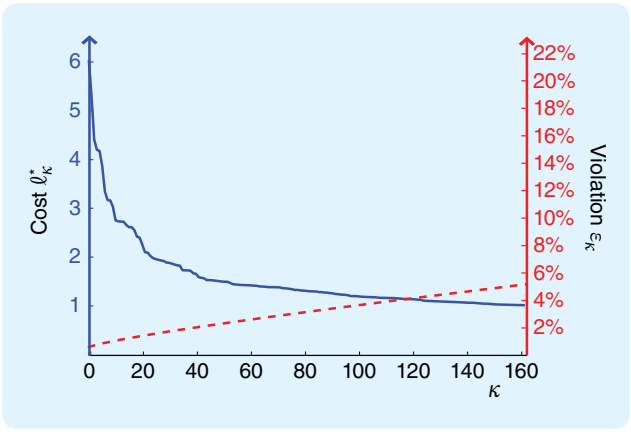


FIGURE 1 Cost-violation plot. In control design in the presence of uncertainty, there is a trade-off between the cost value and the probability with which the cost value is attained. The cost-violation plot is the output generated by the algorithm described in this article. The parameter k indicates the number of scenarios that have been removed, and no longer taken into consideration as constraints, at a certain iteration of the algorithm. A distinct control design is associated with each k. The solid curve represents a probabilistically guaranteed limit on the cost value, and the dashed curve bounds the probability that this limit is exceeded, that is, the dashed curve quantifies the probability that the incurred cost value is actually larger than the value given by the first curve.

This article is in two parts. The first part describes general principles. A broad point of view is taken on approaches that can be used to deal with uncertainty. The second part is concerned with algorithms. Specifically, an algorithm called variable robustness control (VRC) is presented. VRC is applicable to all problems where the cost function is convex in the optimization variables, which covers many situations of practical interest. VRC provides a set of alternative designs, and the user can choose among various options guided by a cost-violation plot that the algorithm also constructs. Figure 1 is an example of a cost-violation plot. Each value of the variable k on the horizontal axis is associated with a distinct design. For each k, the solid curve returns a probabilistically guaranteed limit on the cost value achieved by the corresponding design, and the dashed curve bounds the probability that the limit is exceeded. In other words, the dashed curve quantifies the probability that the incurred cost value is actually larger than the value given by the first curve. The final decision concerning which design to implement rests with the user, who makes a choice that takes into account the problem at hand and his/her attitude to the risk. The cost-violation plot provides a quantitative support to guide the user to make this decision.

The VRC algorithm builds on an established framework for robust optimization based on randomized techniques.

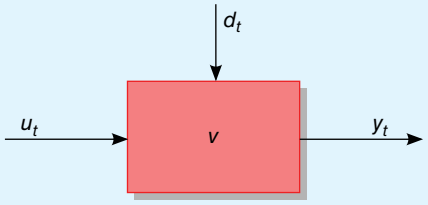


FIGURE 2 An uncertain system. Uncertainty in a system can arise in various ways, through imprecise knowledge of its parameters or structure, or due to the presence of unknown exogenous signals (that is, disturbances). The figure represents a system with an uncertain structural parameter v and affected by the disturbance d_t.

PRINCIPLES

Uncertainty can arise in various ways. *Structural uncertainty* refers to an imprecise knowledge of the system dynamics, whereas *input uncertainty* is associated with unknown exogenous signals, that is, disturbances, that steer the plant away from its nominal behavior (see Figure 2). This article takes a general approach that covers both types of uncertainty. The aim is to provide the user with tools that can be used in a broad range of situations.

Throughout this article, an uncertain element is indicated with the symbol δ, while Δ is the range set for δ. If, for example, the pole p of a stable continuous-time system is uncertain, then $\delta = p$ and Δ is the open left-half complex plane. If instead uncertainty consists of a disturbance $d_t = A\,\text{step}(t - t_0)$ whose amplitude and step time are unknown, then $\delta = (A, t_0)$ and $\Delta = \mathbb{R}^2$.

Define $\theta \in \mathbb{R}^d$ as the vector of design variables; for instance, θ can contain the parameters of a controller or of a state estimator. The cost function to be minimized is written as $\ell(\theta, \delta)$. The simultaneous presence of θ and δ reflects that only partial knowledge of the final optimization result is available through θ because the final result also depends on the uncertainty δ. This setup leads to consider an optimization problem with uncertainty,

$$\text{OPU}: \min_{\theta \in \mathbb{R}^d} \ell(\theta, \delta), \ \delta \in \Delta. \tag{1}$$

As stated in (1), however, the problem is not completely formalized because (1) does not describe how to account for the uncertain element δ when optimization is performed. Addressing this issue requires being more specific about the role of uncertainty, and various approaches arise depending on the adopted formulation.

The Worst-Case Approach

The notion of uncertainty is inescapably linked to the notion of a set, since there cannot be uncertainty without a set of possible uncertainty outcomes. Without any further structure given to the uncertainty other than the uncer-

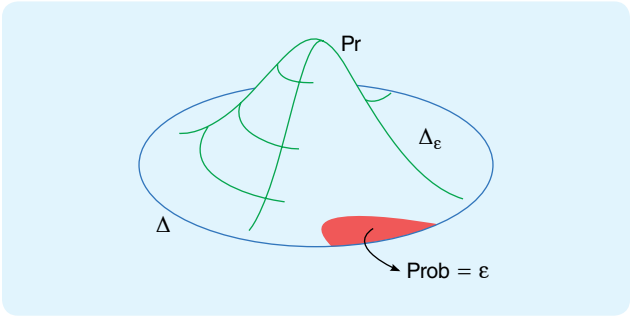


FIGURE 3 A set where the performance is not guaranteed. Δ is the uncertainty set and Pr is the probability with which the various uncertainty outcomes occur. Shown in red is the set where the performance is not guaranteed, and ε is the probability of this set.

tainty ranging in a set Δ, one way to pose the optimization problem is through a worst-case approach

$$\text{OP}-\text{WC}: \min_{\theta\in\mathbb{R}^d}\Big[\max_{\delta\in\Delta}\ell(\theta,\delta)\Big]. \tag{2}$$

In control, the worst-case philosophy was initially adopted out of concerns for stability [1], where H_∞ control was used as the robust alternative to linear-quadratic-Gaussian (LQG) control, which was shown to be potentially very sensitive to small parametric uncertainties [2]. The worst-case approach has become mainstream in robust control, and has been applied to a variety of control problems. Many books are available that describe the worst-case approach in control (e.g., [3]–[5]).

Many methods have been developed for the construction of uncertainty sets. Some of these methods are purely deterministic and are based on the derivation of upper bounds on the set of parameters that are consistent with data, whereas other methods allow the user to express preferences in terms of a coherent risk measure [6] (see [7] and [8] for related contributions).

The Average Approach

When a more structured and probabilistic point of view in the description of uncertainty is adopted, the average approach can be used. In a probabilistic description, Δ is endowed with a probability measure Pr, which has various interpretations depending on the problem at hand. Sometimes Pr describes the chance that various outcomes of the uncertainty element δ occur; other times Pr is a descriptor of the relative importance given to the various uncertainty outcomes. Pr can be used to weigh the δs, to obtain an average cost optimization problem,

$$\text{OP}-\text{A}: \min_{\theta\in\mathbb{R}^d} E_\Delta[\ell(\theta,\delta)] = \min_{\theta\in\mathbb{R}^d}\int_\Delta \ell(\theta,\delta)\,\text{dPr}. \tag{3}$$

This framework is often adopted when uncertainty is associated with disturbance signals [9]–[11], although the average approach can also be used for structural uncertainty [12], [13]. A typical example of the use of the average approach is quadratic stochastic control, where the average cost in discrete time is

$$E_\Delta[\ell(\theta,\delta)] = E_\Delta\left[\sum_{t=0}^{T}\{x_t^T Q x_t + u_t^T R u_t\}\right], \tag{4}$$

where x_t is the system state, u_t is the input, and the expectation is taken with respect to the realizations of the disturbance acting on the system. The δ corresponds to a noise realization, and Δ is the set of all possible noise realizations, whereas the probability Pr indicates the type of noise that is considered. For example, if the noise is white and Gaussian, then Pr is the product probability measure of T independent Gaussian random variables. A celebrated approach to deal with control problems in this framework is dynamic programming (for example, see [11]). However, dynamic programming fails to provide viable methods to deal with constraints, an important feature in many control problems [14]–[17], and [18] presents an alternative based on results from robust optimization.

The average approach is less conservative than the worst-case approach; nevertheless its conservatism can be enhanced by the introduction of an exponential cost. For example, in stochastic control with the definition $J := \sum_{t=0}^{T}\{x_t^T Q x_t + u_t^T R u_t\}$, $E_\Delta[\exp(J)]$ can be used instead of considering the quadratic cost $E_\Delta[J]$. Due to the exponential function, this approach incorporates higher order statistics [19], and the penalty in the occurrence of values of J larger than $E_\Delta[J]$ outweighs the alleviation in the penalty caused by the occurrence of values less than $E_\Delta[J]$, corresponding to a pessimistic viewpoint. A connection between this average-exponential approach and the worst-case approach has been established for linear systems [20], [21] and nonlinear systems [22], [23].

Worst-Case Versus Average Approach

When the uncertainty level is moderate enough that the solution to problem (2) secures an adequate performance for all uncertainty outcomes, the worst-case approach is a suitable design methodology. For other problems the uncertainty is much larger, and the worst-case approach becomes too conservative. This situation may occur for structural uncertainty and is almost invariably true when uncertainty consists of external disturbance signals. In these cases, the average approach can be a suitable alternative. The average approach, however, has a drawback in that it does not allow the user to modulate the robustness level.

The Modulating Robustness Approach

Averaging is not the only way to make use of probability. An alternative approach consists of using probability to quantify the chance that a certain performance specification is attained. This approach has been used in analysis problems in the context of flight control [24] and has been further developed in various directions [25]–[28]. One way to evalu-

ate the probability that a certain performance is attained is through randomization, and explicit bounds on the number of samples needed to quantify the chance are available [29], [30]. Sequential methods can be used to solve feasibility problems with probability specifications [31]–[36].

This article presents a similar approach to that taken in the above references and uses the probability to quantify the chance that a certain performance is attained, but it also considers optimization problems where the level of robustness is modulated.

Precisely, referring to Figure 3, the aim is to minimize the max cost with max taken over a reduced set $\Delta_\varepsilon \subset \Delta$ having probability $\Pr\{\Delta_\varepsilon\} = 1 - \epsilon$, namely,

$$\text{OP} - \text{WC}_\varepsilon : \min_{\theta \in \mathbb{R}^d, \Delta_\varepsilon} \left[\max_{\delta \in \Delta_\varepsilon} \ell(\theta, \delta) \right]. \tag{5}$$

Defining the optimal solution of (5) as θ_ε^* and the optimal value ℓ_ε^*, these terms are related by $\ell_\varepsilon^* = \max_{\delta \in \Delta_\varepsilon} \ell(\theta_\varepsilon^*, \delta)$, that is, ℓ_ε^* is guaranteed against all uncertainty outcomes in Δ_ε which is a set having probability $1 - \varepsilon$.

The reason for leaving out an ε-probability set is to reduce the optimal value as ℓ_ε^* compared with that obtained with the worst-case approach. The level of robustness depends on ε, and, for a given ε, the set Δ_ε in (5) must be selected so that the reduction of the value is maximized. This observation justifies the presence of Δ_ε as an optimization variable in the min quantifier. The minimization with respect to Δ_ε, however, is a very difficult task to achieve in general.

In the approach outlined above, the parameter ε can also be varied and used as a tuning knob. The larger the value for ε, the better the performance, but higher is the risk of performance violation. The level of robustness is adjustable, and so this approach is called a *modulating robustness approach*.

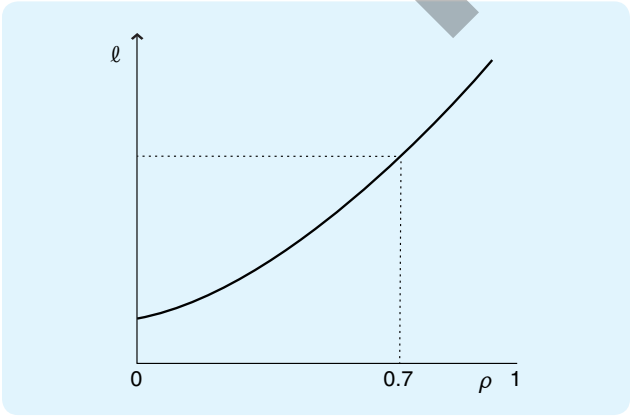


FIGURE 4 Size-cost plot. The plot represents the improvement in the optimal worst-case value ℓ when the uncertainty set is resized by $\rho < 1$. In the figure, a 30% improvement in the optimal worst-case value is achieved by reducing ρ to the value 0.7, which, however, corresponds to an unacceptable loss of robustness.

A Clarifying Remark

To modulate robustness, a straightforward approach consists of resizing the uncertainty set. For example, if Δ is the unitary H_∞ ball, then the ball of radius ρ, $\rho \in [0,1]$, can be considered, and, if Δ is the box $[-1,1]^q$ containing q uncertain variables, then the box $[-\rho, \rho]^q$ is the resized version of Δ. In this approach, the worst-case solution for various values of ρ can be assessed, and someone may think that a satisfactory compromise between performance and robustness can be obtained by selecting a suitable value of ρ. It is a fact, however, that this approach generally fails to provide satisfactory results. What is typically found is a size-cost plot as shown in Figure 4. To improve the optimal worst-case value, ρ has to be decreased to, for example, 0.7 or less, which can correspond to a dramatic drop in terms of robustness. For example, in a box in ten dimensions with uniform probability, $\rho = 0.7$ shrinks the probability from one down to $0.7^{10} = 0.028 < 3\%$, that is, 97% of the probabilistic volume of the original uncertainty set would be left out.

What is interesting is that the same optimal value as for $\rho = 0.7$ is attained in many problems by leaving out a much smaller, more accurately selected, portion of Δ. This portion is hardly ever the outer shell of Δ, which is what the resizing approach discards, and in fact the set Δ_ε has to be determined based on a careful inspection of the optimization problem so as to decide which region of Δ produces the largest improvement of the optimal value if left out. Figure 5 illustrates the situation. In [37], the region to be left out has been described as having an elongated shape to visualize that, typically, this region has a small volume, despite having a significant linear extension, and the term "icicle geometry" has been used to signify this fact.

Finding a suitable region to leave out is a formidable task in general. The VRC algorithm presented in the "Algorithms"

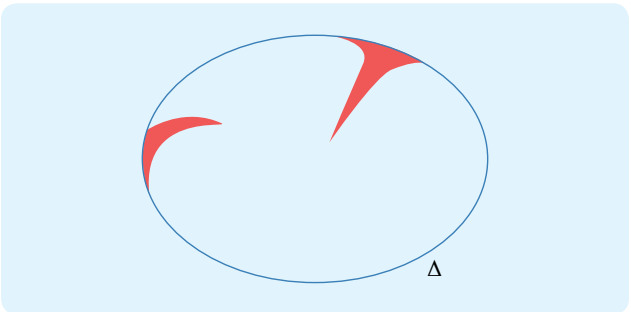


FIGURE 5 The icicle geometry. In typical optimization problems, the cost value can be improved by leaving out a portion of the uncertainty set Δ that has small volume, despite the portion having a significant linear extension. This fact is illustrated in the figure, where the red region is the portion to be left out. This phenomenon is known as "icicle geometry," which finds its explanation in the fact that normally Δ is a set in a high-dimensional space. For example, if Δ is the box $[0,1]^{10}$, and the region to be left out is $[0, \frac{1}{2}]^{10}$, this region stretches from one corner of Δ to its center, although this region only has volume $1/2^{10} < 0.1\%$.

Approaches to Uncertain Optimization: A Visualization in the Optimization Domain

Consider a cost function $\ell(\theta,\delta)$ where $\theta \in \mathbb{R}^d$ is a vector of optimization variables, and δ is an uncertain parameter whose presence expresses the fact that the final result of the optimization procedure is also affected by uncertain elements that do not depend on the user's choices. For one given $\delta, \ell(\theta,\delta)$ is a function of the optimization variable θ only, and a graphical visualization of one such function is provided in Figure S1. As δ is varied, functions $\ell(\theta,\delta)$ form a cloud (see again Figure S1) called the "performance cloud." The various paradigms to uncertain optimization can be described by referring to this cloud.

The worst-case paradigm consists of solving the optimization problem

$$\min_{\theta \in \mathbb{R}^d} \left[\max_{\delta \in \Delta} \ell(\theta,\delta)\right].$$

Referring to Figure S2, the top border of the performance cloud represents the worst-case cost function $\max_{\delta\in\Delta}\ell(\theta,\delta)$ and θ^*_{WC} is its minimizer. This paradigm is pessimistic and thereby conservative.

The average approach requires introducing a probability measure Pr over Δ. Given Pr, the average cost optimization problem is

$$\min_{\theta \in \mathbb{R}^d} \int_{\Delta} \ell(\theta,\delta)\,\mathrm{dPr}.$$

Referring again to Figure S2, this approach corresponds to cutting the performance cloud along the vertical direction originated from every given θ value, and averaging over the δs to determine $\int_{\Delta}\ell(\theta,\delta)\mathrm{dPr}$. As θ changes, $\int_{\Delta}\ell(\theta,\delta)\mathrm{dPr}$ gives the average cost function, whose minimizer $\theta^*_{\text{average}}$ is called the average minimizer.

Finally, the modulating robustness approach consists of solving the problem

$$\min_{\theta \in \mathbb{R}^d, \Delta_\varepsilon} \left[\max_{\delta \in \Delta_\varepsilon} \ell(\theta,\delta)\right],$$

a problem similar to the worst-case approach in which, however, the max requirement is relaxed in a probabilistic sense. Precisely, a max cost is minimized with max taken over a reduced set $\Delta_\varepsilon \subset \Delta$ having probability $\Pr\{\Delta_\varepsilon\} = 1-\varepsilon$. Leaving out the set $\Delta - \Delta_\varepsilon$ improves the performance with respect to the worst-case approach, and the Δ_ε that gives the best improvement is chosen. As ε is varied, various solutions are obtained.

In Figure S3, the curves marked with 1%, 2%, …, called the chance-constrained cost functions, represent, θ by θ, the best possible cost value that can be achieved by leaving out a 1%, 2%, … of the cost functions. Corresponding to each given θ, the best cost value is obtained by moving down from the top border of the cloud until 1%, 2%, … of the cost functions are left above. By minimizing the curve marked with, say, 1%, the minimizer $\theta^*_{1\%}$ is obtained, which carries a risk of 1% that the corresponding optimal value will not be achieved.

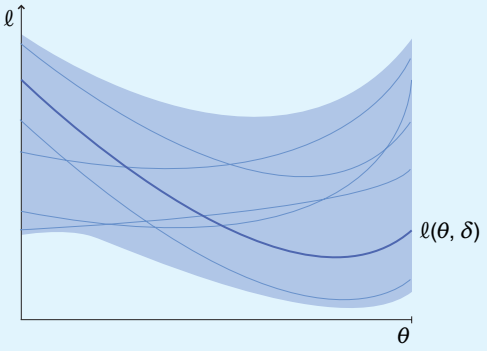


FIGURE S1 The performance cloud. The blue curve represents the cost function $\ell(\theta,\delta)$ for a given outcome of uncertainty δ. As δ is varied, the functions $\ell(\theta,\delta)$ form the performance cloud.

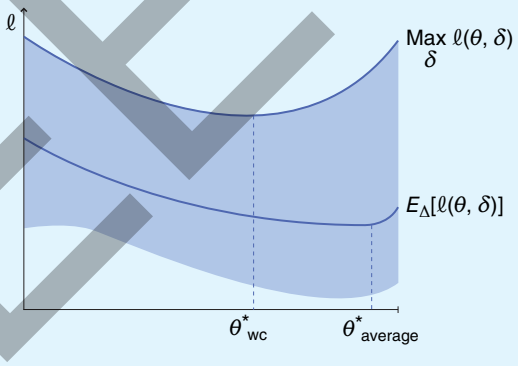


FIGURE S2 Worst-case and average cost functions. The worst-case cost function is the top border of the performance cloud, whereas the average cost function is the curve in the barycentric position.

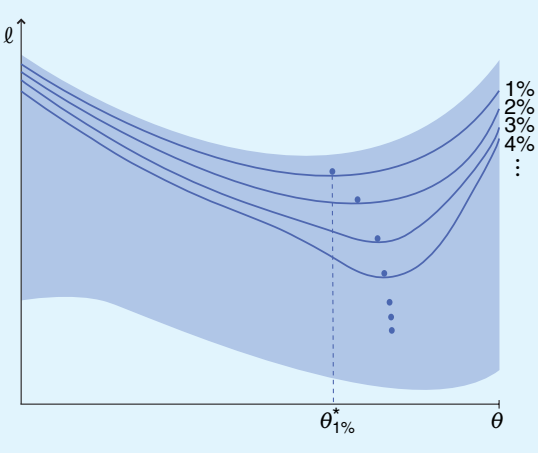


FIGURE S3 Chance-constrained cost functions. The curves marked with 1%, 2%, … are called the chance-constrained cost function and represent, θ by θ, the best possible cost value that can be achieved by leaving out 1%, 2%, … of the cost functions.

Convexity

A function $f(\theta), \theta \in \mathbb{R}^d$, is *convex* if

$$f(\alpha\theta' + (1-\alpha)\theta'') \le \alpha f(\theta') + (1-\alpha) f(\theta'')$$

holds for all θ', θ'', and $\alpha \in [0,1]$. Graphically, convexity of f means that the line segment between every two points on the graph of f lies above the graph of f (see Figure S4).

Convexity is beneficial in optimization because every local minimizer of a convex function is also a global minimizer, and convex functions can therefore be minimized more easily than generic functions, for example, by means of interior point methods.

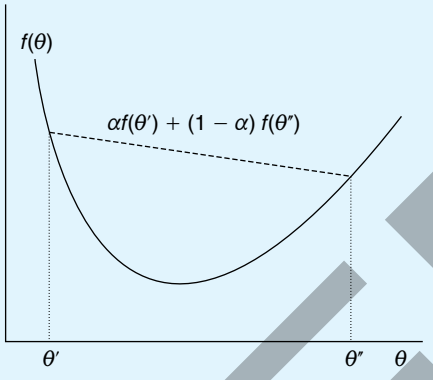


FIGURE S4 A convex function. The line segment between every two points on the graph of a convex function f lies above the graph of f.

section provides a viable route to finding approximate solutions to the modulating robustness approach.

Some Bibliographical Notes

The above approach of minimizing a cost function over an uncertainty set of reduced probability $1-\varepsilon$ has a long history in optimization theory, where the approach is known as "chance-constrained" optimization. This approach dates back at least to the 1950s [38], and countless contributions have appeared in the literature; see [39]–[44] for more information on the history of this method in optimization.

In systems and control, this approach is newer and has been considered in fewer papers [45]–[57]. We believe that this different degree of emphasis is due to two distinct reasons. One reason is tradition. When dealing with stochastic disturbances, the approach commonly adopted in the control literature is the average cost method [9], [10]. For structural uncertainty, the origin of robust control was a concern for stability, for which a probabilistic compromise was not considered acceptable. Following this origin, robustness and worst-case analysis have traveled hand in hand in the robust control community for more than 30 years, e.g., see [3]–[5]. The second reason for relatively low emphasis on chance-constrained optimization in the control literature is that the approach (5) still lacks suitable algorithmic methods to find solutions. The VRC algorithm in the "Algorithms" section describes how to fill this gap.

"Approaches to Uncertain Optimization: A Visualization in the Optimization Domain" provides more comparisons among the various approaches to uncertain optimization described in this section.

ALGORITHMS

Here the VRC algorithm for the implementation of the modulating robustness approach is presented. The main idea behind VRC is to replace the infinite set Δ with a finite approximant obtained by a randomization process over Δ, and to further remove samples belonging to this approximant so as to improve the optimal value. The algorithm and its theoretical properties are derived under the assumption of convexity (see "Convexity").

Assumption 1 (Convexity)

For every δ, the function $\ell(\theta, \delta)$ is convex with respect to θ.

In contrast, the dependence of $\ell(\theta, \delta)$ on δ is totally arbitrary. The fact that $\ell(\theta, \delta)$ is convex in θ is a restrictive assumption, which is nevertheless satisfied in many problems. Moreover, many problems that are not convex in their natural formulation are amenable to convex reformulation, for example, by means of linear matrix inequalities (LMIs); see "Linear Matrix Inequalities" for details on LMIs and relevant references.

An additional assumption is introduced to streamline the presentation.

Assumption 2

The solution to every min-max problem

$$\min_{\theta \in \mathbb{R}^d} \left[\max_{\delta \in F} \ell(\theta, \delta) \right], \tag{6}$$

where max is taken over a finite set $F = \{\delta_1, \delta_2, \ldots, \delta_p\} \subseteq \Delta$, exists and is unique.

The VRC algorithm builds on the scenario approach of [43] and [58]–[60]. This article considers removal of constraints to trade robustness for performance, following an approach whose foundations have been settled in [57].

The Variable Robustness Control Algorithm

Let $\delta_1, \delta_2, \ldots, \delta_N$ be N uncertainty outcomes, hereafter called *scenarios*, extracted independently of each other from Δ according to the probability measure Pr. References [61]–[63] provide algorithms to perform random extractions for various Pr and Δ, and [64] gives a general overview of randomized methods. This set of scenarios is used as a *surrogate* or

Linear Matrix Inequalities

A linear matrix inequality (LMI), [S1], [S2], is an expression of the form

$$F_0 + F_1\theta_1 + F_2\theta_2 + \cdots + F_d\theta_d \preccurlyeq 0, \qquad \text{(S1)}$$

where $\theta_i, i = 1,\ldots,d$, are real variables, $F_i, i = 0,1,\ldots,d$, are $n \times n$ symmetric matrices, and $A \preccurlyeq 0$ means that A is a negative-semidefinite matrix, that is, $z^T A z \leq 0$ for all vectors $z \in \mathbb{R}^n$. The set of θ where (S1) is satisfied is a convex set, that is, an LMI specifies a convex constraint on

$$\theta = [\theta_1\ \theta_2 \cdots \theta_d]^T.$$

LMIs are useful tools to describe constraints arising in systems and control applications [S2]– [S7], while the use of LMIs is also fostered by the existence of software for the numerical solution of optimization problems with LMI constraints such as CVX [S8], [S9], or YALMIP [S10].

REFERENCES

[S1] S. Boyd and V. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[S2] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM Press, 1994.

[S3] P. Gahinet, "Explicit controller formulas for LMI-based H_∞ synthesis," *Automatica*, vol. 32, no. 7, pp. 1007–1014, 1996.

[S4] P. Apkarian and H. D. Tuan, "Parameterized LMIs in control theory," *SIAM J. Control Optim.*, vol. 38, no. 4, pp. 1241–1264, 2000.

[S5] P. Apkarian, H. D. Tuan, and J. Bernussou, "Continuous-time analysis, eigenstructure assignment, and H_2 synthesis with enhanced linear matrix inequalities (LMI) characterizations," *IEEE Trans. Autom. Contr.*, vol. 46, no. 12, pp. 1941–1946, 2001.

[S6] C. W. Scherer, "Relaxations for robust linear matrix inequality problems with verifications for exactness," *SIAM J. Matrix Anal. Appl.*, vol. 27, no. 2, pp. 365–395, 2005.

[S7] C. W. Scherer, "LMI relaxations in robust control," *Eur. J. Control*, vol. 12, no. 1, pp. 3–29, 2006.

[S8] M. Grant and S. Boyd. (2010, Apr.). CVX: Matlab software for disciplined convex programming, version. 1.21. [Online]. Available: http://cvxr.com/cvx

[S9] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, Eds. London, U.K.: Springer-Verlag, 2008.

[S10] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in Matlab," in *Proc. 62 CACSD Conf.*, 2004, pp. 284–289.

a *descriptor* of Δ and is useful in the development of practical algorithms because it only contains finitely many elements. As it will be shown, the VRC procedure, which only depends on $\delta_1, \delta_2, \ldots, \delta_N$, is backed by precise robustness results related to the whole set Δ, a fact stated in Theorem 1 below.

First a procedure is presented for computing θ_k^* and ℓ_k^*, that is, the optimal design parameters and optimal values obtained after the removal of an increasing number k of scenarios. This procedure is instrumental to the derivation of the complete VRC algorithm, which is subsequently presented.

TABLE 1 Procedure for computing θ_k^* and ℓ_k^* for $k = 0, 1, \ldots, \bar{k}$.

1: set $G_0 = \{1, 2, \ldots, N\}$;
solve the program

$$\min_{\theta \in \mathbb{R}^d} \left[\max_{i \in G_0} \ell(\theta, \delta_i) \right];$$

let θ_0^* be the optimal solution and $\ell_0^* = \max_{i \in G_0} \ell(\theta_0^*, \delta_i)$ be the optimal value;
set $Z = G_0$, $\widehat{\theta} = \theta_0^*$, and $\hat{\ell} = \ell_0^*$;

2: FOR $k := 1$ TO $\bar{k}$

2.1: set $A = \{j \in Z : \ell(\widehat{\theta}, \delta_j) = \hat{\ell}\}$; % (*A contains the active scenarios*)

2.2: FOR $j \in A$

solve the program

$$\min_{\theta \in \mathbb{R}^d} \left[\max_{i \in Z - j} \ell(\theta, \delta_i) \right];$$

let $\widehat{\theta}^j$ be the optimal solution and $\hat{\ell}^j = \max_{i \in Z - j} \ell(\widehat{\theta}^j, \delta_i)$ be the optimal value;

IF $\hat{\ell}^j < \hat{\ell}$ THEN set $\widehat{\theta} = \widehat{\theta}^j$ and $\hat{\ell} = \hat{\ell}^j$

END FOR

2.3: set $Z = \{i \in \{1, 2, \ldots, N\} : \ell(\widehat{\theta}, \delta_i) \leq \hat{\ell}\}$;

2.4: IF $|Z| > N - k$ THEN GOTO 2.1 % ($|Z|$ = cardinality of set Z)

ELSE set $G_k = Z$, $\theta_k^* = \widehat{\theta}$ and $\ell_k^* = \hat{\ell}$;

END FOR

Procedure for Computing θ_k^* and ℓ_k^*

Replacing Δ with $\delta_1, \delta_2, \ldots, \delta_N$ in the worst-case approach (2) yields

$$\min_{\theta \in \mathbb{R}^d} \left[\max_{i=1,\ldots,N} \ell(\theta, \delta_i) \right]. \qquad (7)$$

The solution θ_0^* of this problem is the starting point of the procedure.

The procedure executes a progressive elimination of scenarios according to a greedy logic and finds the optimal solutions $\theta_1^*, \theta_2^*, \ldots, \theta_k^*, \ldots$ obtained after the elimination of these scenarios. The variable θ_k^* is called the *design at level k*. To be specific, let G_k be the set of scenarios that have survived after the elimination of k scenarios, and let θ_k^* be the solution to the optimization problem,

$$\min_{\theta \in \mathbb{R}^d} \left[\max_{i \in G_k} \ell(\theta, \delta_i) \right], \qquad (8)$$

and $\ell_k^* := \max_{i \in G_k} \ell(\theta_k^*, \delta_i)$ be the corresponding optimal value. To update G_k, it has to be decided which scenario has to be removed next. To this aim, the procedure scans, one by one, the scenarios in G_k and selects the scenario that, if

removed, gives the largest improvement in the optimal value. Eliminating this scenario from G_k gives G_{k+1}.

The pseudocode in Table 1 implements the above scheme.

Since max is taken in points 1 and 2.2 with respect to a finite number of scenarios, these problems can be solved by means of standard optimization programs, such as the openly distributed CVX [65], [66] or YALMIP [67]. The FOR cycle in 2.2 removes, one by one, scenarios in search of the scenario whose elimination gives the largest improvement in the optimal value. The search is restricted to the δ_j such that $\ell(\widehat{\theta}, \delta_j) = \hat{\ell}$, that is, the active scenarios, since the elimination of nonactive scenarios cannot possibly improve the optimal value. Upon exiting the FOR cycle in 2.2, the minimizer is stored in $\widehat{\theta}$ and the corresponding optimal value is $\hat{\ell}$. The procedure, however, does not yet update G_k because of a detail that requires a bit of explanation. The fact is that $\widehat{\theta}$ is only a potential solution at level k since removing one scenario δ_j can generate a pair $(\widehat{\theta}, \hat{\ell})$ such that $\ell(\widehat{\theta}, \delta_i) \leq \hat{\ell}$ for some previously removed scenario δ_i. If so, this δ_i is reinstated and the procedure eliminates another scenario before outputting the design θ_k^* at level k, and point 2.4 executes a test to decide whether or not this situation has occurred. The outer FOR in point 2 cycles over the k values, and point 1 is an initialization.

An inspection of the pseudocode reveals that the condition for the procedure to come to termination is that, every time the FOR cycle in 2.2 is run, an active scenario is found whose elimination improves the optimal value. This fact is generally true, and termination with probability one is assumed to hold throughout the sequel.

The procedure for computing θ_k^* and ℓ_k^* implements the idea of discarding scenarios to improve the optimal value. Scenario discarding is implemented according to a greedy logic that makes the procedure computationally feasible.

The procedure operates over a finite set of scenarios $\delta_1, \delta_2, \ldots, \delta_N$, and the sole conclusion that can be drawn at the present stage is that ℓ_k^* bounds the cost $\ell(\theta_k^*, \delta)$ for all $\delta \in G_k$. The scenarios $\delta_1, \delta_2, \ldots, \delta_N$ are the visible uncertainty outcomes, that is, those scenarios that the procedure uses. A question arises as to whether the optimal value ℓ_k^* is also a limit to the cost value achieved in correspondence of the other δs, that is, those scenarios that have not been seen. This is a generalization question, and it is concerned with inferring the "invisible" from the "visible," that is, the unseen δs from the seen scenarios. This question is addressed in a precise manner in Theorem 1 after formalizing the VRC algorithm.

The Variable Robustness Control Algorithm

The following definition formalizes the concept of cost violation probability.

Definition 1 (Cost Violation Probability)

Given an instance of the design variable $\bar{\theta}$ and a cost $\bar{\ell}$, the *cost violation probability* of $(\bar{\theta}, \bar{\ell})$ is defined as

$$V(\bar{\theta}, \bar{\ell}) := \Pr\{\delta \in \Delta : \ell(\bar{\theta}, \delta) > \bar{\ell}\}, \tag{9}$$

that is, $V(\bar{\theta}, \bar{\ell})$ is the probability with which the cost value obtained with $\bar{\theta}$ is larger than $\bar{\ell}$.

$V(\bar{\theta}, \bar{\ell})$ quantifies the level of robustness of $\bar{\ell}$ relative to the whole uncertain set Δ when the design variable is $\bar{\theta}$. The VRC algorithm outputs θ_k^* and ℓ_k^*, and also returns ε_k, which is an evaluation of $V(\theta_k^*, \ell_k^*)$. Thus, VRC complements the design with probabilistic performance guarantees, by means of which a plot such as the plot shown in Figure 1 is constructed. When using VRC, the user inspects the numerical values ℓ_k^* against the probabilities ε_k to select a design θ_k^* that meets a suitable compromise.

The algorithm has three inputs: $\bar{\varepsilon}$, $\alpha\%$, and β. The input $\bar{\varepsilon}$ is an upper bound for ε_0, which is the cost violation probability for $k = 0$. The input $\alpha\%$ is the proportion of the total number N of scenarios that are discarded upon exiting the algorithm. The input β is a confidence parameter that requires a bit of additional explanation. Due to randomization, θ_k^* and ℓ_k^* are random elements that depend on $\delta_1, \delta_2, \ldots, \delta_N$ and so is the result that $V(\theta_k^*, \ell_k^*) \leq \varepsilon_k$. Hence, the probability that $V(\theta_k^*, \ell_k^*)$ is larger than ε_k can be made small but this event cannot be totally excluded. The input β allows the user to exercise the option to reduce the probability that $V(\theta_k^*, \ell_k^*) > \varepsilon_k$ below a desired level β. From a practical point of view, the parameter β has a minor importance because selecting a value for β so small that β becomes negligible, for example $\beta = 10^{-7}$, affects only marginally the computational burden of the algorithm. This issue is discussed in detail in the section "Further Comments on N and ε_k." That section also shows that the evaluation ε_k is tight. The pseudocode of the VRC algorithm is given in Table 2.

The computation of the value of N in point 1 of the algorithm can be performed by means of the betainc function of Matlab. For the benefit of practitioners, a complete code is provided in "Matlab Code to Compute N." "Matlab Code to Compute ε_k" provides a Matlab code for computing the ε_k in point 4.

The fact that ε_k bounds $V(\theta_k^*, \ell_k^*)$ is established in the next theorem.

Theorem 1

The relation $V(\theta_k^*, \ell_k^*) \leq \varepsilon_k$ holds true simultaneously for all $k = 0, 1, \ldots, \overline{k}$ with probability of at least $1 - \beta$.

The theorem holds for any probability Pr. Its proof rests on establishing the result that the probability that $V(\theta_k^*, \ell_k^*) > \varepsilon_k$ is bounded by the left-hand side of (11) in Table 2. This quantity is set equal to $\beta / (\overline{k} + 1)$ in (11), so that $V(\theta_k^*, \ell_k^*) \leq \varepsilon_k$ holds for all $k = 0, 1, \ldots, \overline{k}$ with probability of

TABLE 2 VRC algorithm.

Input: $\bar{\varepsilon}, \alpha\%, \beta$
Output: $\theta_k^*, \mathbb{l}_k^*, \varepsilon_k$

1: compute the smallest integer $N \geq d$ (recall that d is the size of θ) such that

$$\sum_{i=0}^{d} \binom{N}{i} \bar{\varepsilon}^i (1-\bar{\varepsilon})^{N-i} \leq \frac{\beta}{\alpha\% \cdot N + 1}; \tag{10}$$

let $\overline{k} := \lfloor \alpha\% \cdot N \rfloor$, which is the integer part of $\alpha\% \cdot N$;

2: sample N independent scenarios $\delta_1, \delta_2, \ldots, \delta_N$ from Δ according to the probability measure Pr;
3: run the "Procedure for computing θ_k^* and $\mathbb{l}_k^*$ for $k = 0, 1, \ldots, \overline{k}$":
RETURN θ_k^* and $\mathbb{l}_k^*$ for $k = 0, 1, \ldots, \overline{k}$;
4: for $k = 0, 1, \ldots, \overline{k}$, solve for ε_k the equation

$$\binom{d+k}{k} \sum_{i=0}^{d+k} \binom{N}{i} \varepsilon_i^k (1-\varepsilon_k)^{N-i} = \frac{\beta}{\overline{k}+1}; \tag{11}$$

RETURN ε_k for $k = 0, 1, \ldots, \overline{k}$.

Matlab Code to Compute *N*

Function inputs: `eps` = $\bar{\varepsilon}$; `alph` = $\alpha\%$; `bet` = β; `d` = number of design variables d.

Note: in the function, `N` is computed by bisection; `N1` is the initial lower bound, while `N2` is the initial upper bound and corresponds to formula (12).

Function findN

```
function N = findN(eps,alph,bet,d)
N1 = d;
N2 = floor(2/eps*(d+log(1/bet)+1)
     + 4/eps*log(2/eps*(d+log
        (1/bet)+1))) + 1;
while N2-N1>1
  N = floor((N1+N2)/2);
  if betainc(1-eps,N-d,d+1)
     > bet/(alph*N+1)
  N1=N;
  else
  N2=N;
  end
end
N = N2;
```

at least $1 - \sum_{k=0}^{\overline{k}} \beta/(\overline{k}+1) = 1 - \beta$. The technical proof of this result and all other results in this article are given in the section "Proofs."

Further Comments on N and ε_k

Sample Complexity *N*

The size N of the scenario set is a significant factor in determining the computational complexity of the VRC algorithm. In fact, the number of times point 3 in the VRC algorithm is run is proportional to N, and, moreover, N impacts the computational complexity of the programs to be solved each time point 3 is run. As shown in the section "Proofs," the N given in point 1 of the VRC algorithm is bounded as

$$N \leq \left\lfloor \frac{2}{\bar{\varepsilon}} \left(d + \ln \frac{1}{\beta} + 1 \right) + \frac{4}{\bar{\varepsilon}} \ln \left(\frac{2}{\bar{\varepsilon}} \left(d + \ln \frac{1}{\beta} + 1 \right) \right) \right\rfloor + 1, \tag{12}$$

which is valid for all values of parameter $\alpha\%$. Equation (12) exhibits an approximately linear dependence on $1/\bar{\varepsilon}$ and a logarithmic dependence on $1/\beta$. Thus, β can be made very small, such as $\beta = 10^{-7}$, with no significant increase in the computational complexity.

The design problem to which VRC is applied enters (12) only through d, the size of θ. The dependence on d is linear. The value of N does not depend at all on the uncertainty set Δ. This result is in contrast with approximation schemes based on a gridding of Δ, where the number of grid points increases exponentially fast with the dimension of the space to which Δ belongs. Thus, the VRC algorithm may offer a viable approach in problems where using a gridding scheme is impractical.

Cost Violation Parameter ε_k

The cost violation parameter ε_k given by point 4 of the VRC algorithm can be bounded as

$$\varepsilon_k \leq \frac{k}{N} + \frac{d + h + \sqrt{h^2 + 2(d+k)h}}{N}, \tag{13}$$

where

$$h = \ln(\overline{k}+1) + \ln \frac{1}{\beta} + d \cdot \left[1 + \ln \frac{d+k}{d} \right], \tag{14}$$

where the inequality is proven in the section "Proofs."

Inequality (13) reveals some interesting features of ε_k. The first term in the bound is k/N, which is the empirical violation of the solution at level k. Due to stochastic fluctuation, we cannot expect that the real violation is bounded by k/N with high confidence $1 - \beta$, and the second term accounts for the gap. For k proportional to N, say $k = \gamma N$, as N increases the second term goes to zero approximately as $O(1/\sqrt{N})$, so that ε_k approaches γ as N grows. Figure

Matlab Code to Compute ε_k

Function inputs: k = number of removed scenarios k; `N` = number of scenarios; `kmax` = $\overline{k}$; `bet` = β; `d` = number of design variables d.

Function findepsk

```
function epsk = findepsk(k,N,kmax,bet,d)
eps1 = 0;
eps2 = 1;
coeff = 1/(d*beta(k+1,d));
while eps2-eps1 > 1e-10
   epsk = (eps1+eps2)/2;
   if coeff*betainc(1-epsk,N-k-d,d+k+1)
      > bet/(kmax+1)
   eps1 = epsk;
   else
   eps2 = epsk;
   end
end
epsk = eps2;
```

6 plots the relationship between ε_k and N for $k = 0.2N$, $d = 2$, $\beta = 10^{-7}$, and $\overline{k} = 0.2N$.

A Simplified and Easier to Implement Version of the Variable Robustness Control Algorithm

This section presents a simplified version of the VRC algorithm that requires less intervention by the user, while losing little in terms of performance.

A first simplification is obtained by providing explicit values for N and ε_k. An inspection of the original VRC algorithm reveals that the result in Theorem 1 continues to hold if VRC is modified in the following two ways. In point 1 of the VRC algorithm, any $N \geq d$ that satisfies inequality (10), not necessarily the smallest N, can be selected. In point 4 of the VRC algorithm, every value larger than that given by the solution of (11) can be attributed to ε_k. Choosing N not to be the smallest and ε_k larger than that given by (11) implies that a price is paid, both in terms of an increased computational complexity, because N is larger than required, and in terms of probabilistic guarantees, because ε_k becomes looser. The simplified VRC algorithm uses the choices of N and ε_k given by the right-hand side of (12) and (13).

To further streamline the usage of the algorithm, β is omitted from the set of inputs assigned by the user. This choice is motivated by the fact that β has a minor impact on the computational complexity of the algorithm and so can be set to a fixed low value. The simplified algorithm takes $\beta = 10^{-7}$, which is small enough to be negligible for most practical purposes.

The algorithm in Table 3 implements the above simplifications. The inputs $\bar{\varepsilon}$ and $\alpha\%$ and all outputs have the same meaning as in the original VRC algorithm.

A SIMULATION EXAMPLE

Consider the autoregressive moving average (ARMA) system

$$y_{t+1} = ay_t + bu_t + c_1 w_t + c_2 w_{t-1}, \tag{15}$$

where u_t and y_t are the input and output, and w_t is white noise with zero mean and unit variance $WN(0,1)$. The parameters a, b, c_1, and c_2 are real, with imprecisely known values that satisfy the stability condition $|a| < 1$ and controllability condition $b \neq 0$.

The noise w_t is measured and the objective is to design a feedforward compensator with structure

$$u_t = \theta_1 w_t + \theta_2 w_{t-1} \tag{16}$$

that minimizes the asymptotic variance of y_t (see Figure 7).

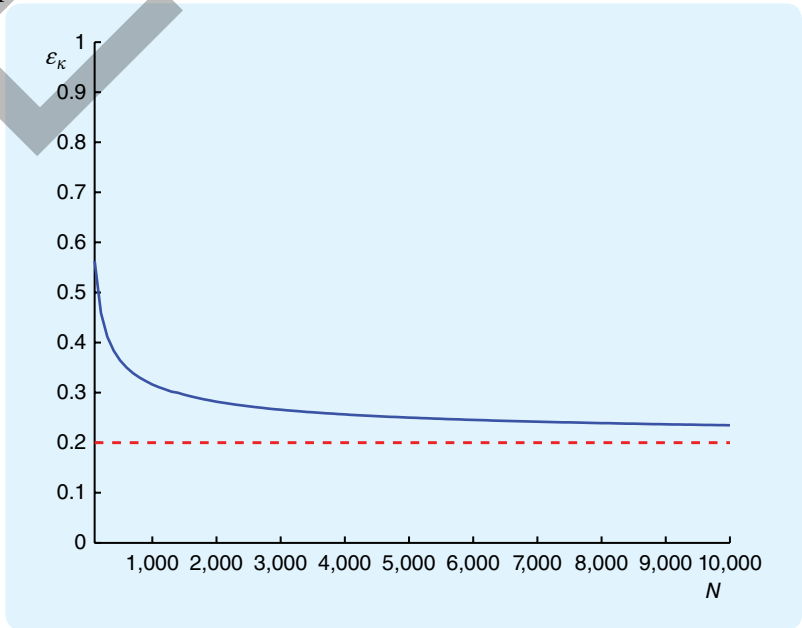


FIGURE 6 Violation versus the number of scenarios. The probability that the cost value found through the variable robustness control algorithm is exceeded is bounded by ε_k, which is plotted as a function of N for $k = 0.2N$, $d = 2$, $\beta = 10^{-7}$, and $\overline{k} = 0.2N$. The plot shows that ε_k approaches the empirical violation k/N as $N \to \infty$.

TABLE 3 VRC—Easy to implement algorithm.

Input: $\bar{\varepsilon}, \alpha\%$
Output: $\theta_k^*, \mathbb{l}_k^*, \varepsilon_k$

1: compute

$$N = \left\lfloor \frac{2}{\bar{\varepsilon}}(d+17.2) + \frac{4}{\bar{\varepsilon}} \ln\left(\frac{2}{\bar{\varepsilon}}(d+17.2)\right) \right\rfloor + 1,$$

$$\bar{k} = \lfloor \alpha\% \cdot N \rfloor; \qquad (\%\lfloor \cdot \rfloor = \text{integer part})$$

2: sample N independent scenarios $\delta_1, \delta_2, \ldots, \delta_N$ from Δ according to the probability measure Pr;
3: run the "Procedure for computing θ_k^* and $\mathbb{l}_k^*$ for $k = 0, 1, \ldots, \bar{k}$";
RETURN θ_k^* and $\mathbb{l}_k^*$ for $k = 0, 1, \ldots, \bar{k}$;
4: for $k = 0, 1, \ldots, \bar{k}$, compute

$$\varepsilon_k = \frac{k}{N} + \frac{d + h + \sqrt{h^2 + 2(d+k)h}}{N},$$

where

$$h = \ln(\bar{k}+1) + 16.2 + d\left[1 + \ln\frac{d+k}{d}\right];$$

RETURN ε_k for $k = 0, 1, \ldots, \bar{k}$.

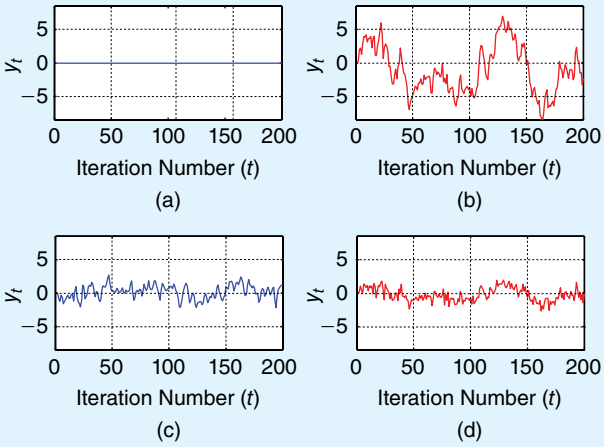


FIGURE 8 The behavior of the nominal compensator and that of the compensator designed with the variable robustness control algorithm for $k = 60$. Plot (a) shows the behavior of the nominal compensator when it is connected to the nominal system, whereas plot (b) is the output when the nominal compensator is connected to a perturbed system. A comparison of (a) and (b) shows the deterioration of performance in the nominal compensator due to changed operating conditions. Plot (c) shows the behavior of the compensator designed with the variable robustness control algorithm for $k = 60$ when connected to the nominal system. Connecting the same compensator to a perturbed system produces the output in plot (d) with no appreciable deterioration in the performance.

If the system parameters a, b, c_1, and c_2 were known, an optimal compensator can be easily found. Indeed, substituting $u_t = \theta_1 w_t + \theta_2 w_{t-1}$ into (15) gives

$$y_{t+1} = a y_t + (c_1 + b\theta_1) w_t + (c_2 + b\theta_2) w_{t-1}, \tag{17}$$

from which the expression for the asymptotic variance of y_t is computed as

$$E[y_t^2] = \frac{(c_1 + b\theta_1)^2 + (c_2 + b\theta_2)^2 + 2a(c_1 + b\theta_1)(c_2 + b\theta_2)}{1 - a^2}. \tag{18}$$

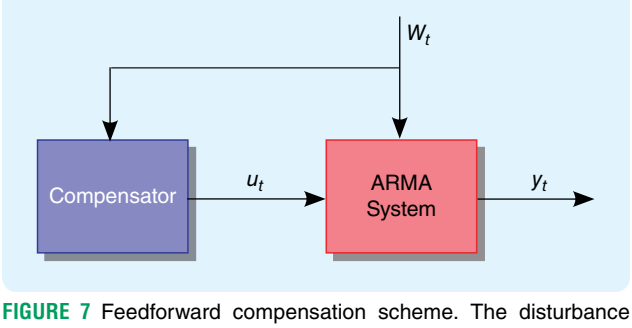


FIGURE 7 Feedforward compensation scheme. The disturbance source w_t is measured and processed by the compensation unit that generates the control action u_t with the objective of minimizing the effect of w_t on y_t.

Hence, the values of θ_1 and θ_2 minimizing $E[y_t^2]$ are

$$\theta_1 = -\frac{c_1}{b}, \quad \theta_2 = -\frac{c_2}{b}, \tag{19}$$

resulting in $E[y_t^2] = 0$.

On the other hand, the system parameter values are not always available in practical situations. More realistically, the parameters are only partially known, with values taken from a given uncertainty set Δ. In this context, the choice of the compensator parameters θ_1 and θ_2 has to be made while taking into account the various dynamical behaviors possible for the system.

As an example, suppose that δ has two components σ_1 and σ_2 both ranging in $[-1, 1]$, that is, $\delta = (\sigma_1, \sigma_2)$ and $\Delta = [-1, 1]^2$, and that the system parameters are expressed as

$$a = \frac{3.5\sigma_1^2 - 0.2}{3\sigma_1^2 + 0.3}(0.32\sigma_1 + 0.6), \tag{20}$$

$$b = 1 + \frac{\sigma_1 \sigma_2^2}{10}, \tag{21}$$

$$c_1 = \frac{-0.01 + (\sigma_1 + \sigma_2^2)^2}{0.02 + (\sigma_1 + \sigma_2^2)^2}\left(1 - \frac{(\sigma_1 - 1)(\sigma_2 - 1)}{2}\right), \tag{22}$$

$$c_2 = \frac{0.05}{0.025 + (\sigma_1 + \sigma_2 - 2)^2}. \tag{23}$$

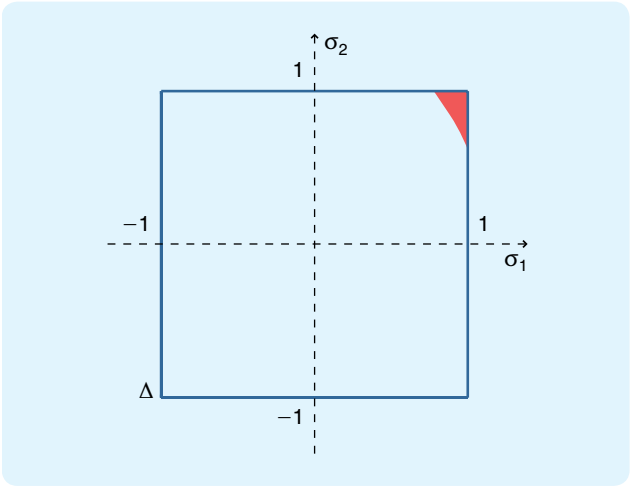


FIGURE 9 The set where the performance is not guaranteed in the compensator example. The variable robustness control algorithm does not guarantee the performance over the whole uncertainty domain Δ, and the red triangle at the top right corner corresponds to plants that do not meet the compensation performance.

The nominal values for σ_1 and σ_2 are $\sigma_1^{\text{nom}} = 0$ and $\sigma_2^{\text{nom}} = 0$, which corresponds to $a^{\text{nom}} = -0.4$, $b^{\text{nom}} = 1$, $c_1^{\text{nom}} = -0.25$, and $c_2^{\text{nom}} = 0.0124$. From (19), the resulting nominal compensator has parameters $\theta_1^{\text{nom}} = 0.25$ and $\theta_2^{\text{nom}} = -0.0124$. Figure 8(a)–(b) shows the output that is obtained when this compensator is connected to the nominal system and to a perturbed system picked at random in the uncertainty domain. The deterioration in performance for the perturbed system does not come as a surprise because the nominal compensator is conceived with no concern for uncertainty.

Now set $\bar{\varepsilon} = 0.5\%$, $\alpha\% = 3\%$, $\beta = 10^{-7}$, and run the VRC algorithm where $\ell(\theta, \delta) = E[y_t^2]$, and Pr is uniform over $[-1, 1]^2$. The resulting N is 5427 and the obtained cost-violation plot is in Figure 1. For each value of k, the output variance ℓ_k^* and the probability ε_k that this output variance can be exceeded are shown.

Based on an inspection of the curves, a reasonable selection for k is 60, where the cost is relatively flat and the probability of violation is low (this choice is subjective and others may opt for a different choice). With this choice, $\varepsilon_{60} = 2.5\%$ and $\ell_{60}^* = 1.42$, with an improvement of 76% over the initial optimal value of 6.00 obtained for $k = 0$. The compensator parameters are $\theta_{1,60}^* = -0.24$ and $\theta_{2,60}^* = -0.59$. According to Theorem 1, with probability $1 - \beta = 1 - 10^{-7}$, which is nearly probability 1, the compensator $u_t = -0.24w_t - 0.59w_{t-1}$ guarantees that $E[y_t^2] \le 1.42$ for all plants in the uncertainty set Δ except for a fraction of plants no more than $\varepsilon_{60} = 2.5\%$. Figure 9 shows the region in the Δ domain where the output variance is larger than 1.42. The volume of the region is 1.2% of the total volume of the uncertainty domain $[-1, 1]^2$, which is below the threshold of 2.5%.

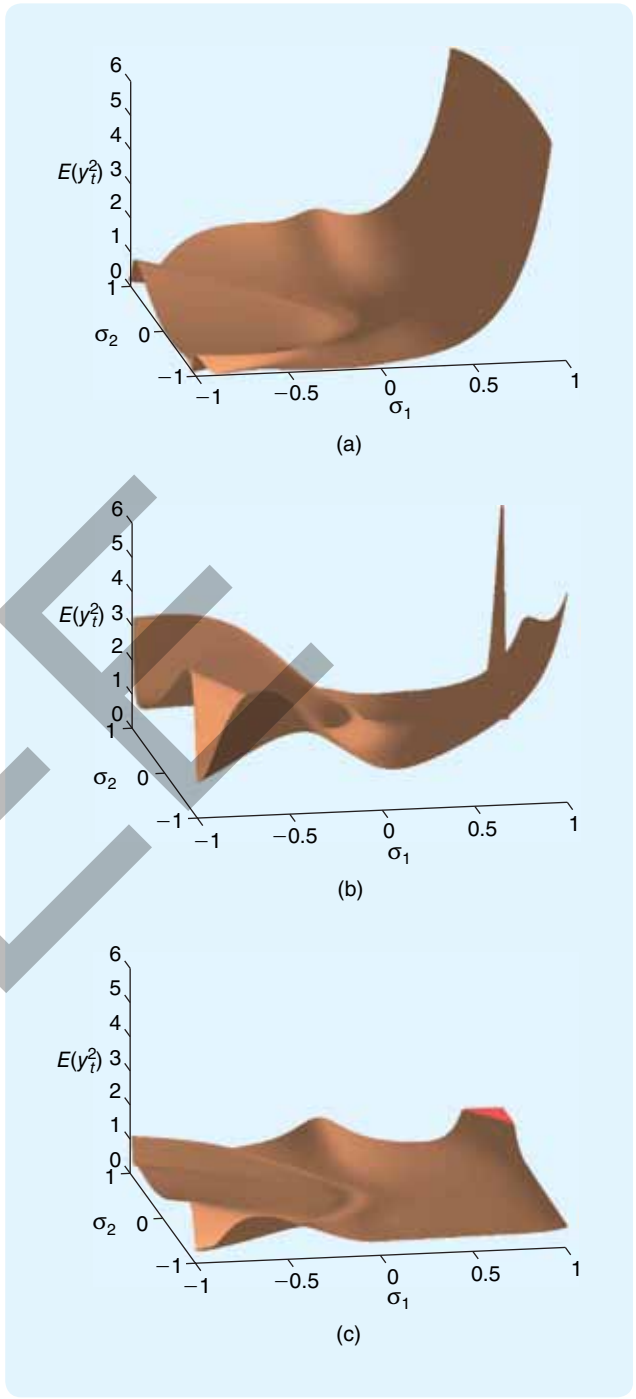


FIGURE 10 Output variance for the (a) nominal compensator, (b) compensator designed with the variable robustness control (VRC) algorithm for $k = 0$, and (c) compensator designed with VRC for $k = 60$. In (a), the values of $E[y_t^2]$ for σ_1 close to one are above six and have been cut off to allow a better visualization. The nominal compensator is designed without any concern for uncertainty, and the output variance is, for many plants, larger than the output variance shown in (c). The compensator designed with VRC for $k = 0$ [see (b)] enforces a high degree of robustness and the whole graph is below the value six, so that no cutting of the graph has been necessary. Still, for many plants the output variance in (b) is larger than in case (c). Allowing for some probability of violation as in (c) improves the overall output variance profile.

In summary, the VRC algorithm has determined a region of small volume whose elimination guarantees a large improvement in the optimal value. This result is achieved by letting the problem speak through the uncertainty samples that have been extracted, whereas an a priori choice of Δ_ε obtained by a resizing of the Δ domain as discussed in the section "A Clarifying Remark" produces little benefit. In this problem, for example, resizing $\Delta = [-1,1]^2$ to $[-0.9874, 0.9874]^2$, which leaves out 2.5% of the total volume, yields a compensator with a optimal value of 5.64. Likewise, leaving out 0.5, 1, 1.5, and 2% of the total volume gives an optimal value of 6.00, 3.03, 1.98, 1.65 with VRC versus 6.20, 6.16, 5.89, 5.81 with the resizing method.

Figure 10 depicts the value of $E[y_f^2]$ achieved for the various systems in Δ by the nominal compensator and by the compensators obtained by the VRC algorithm for $k = 0$ and $k = 60$. In Figure 10(c), the flat zone close to the corner (1, 1) corresponds to the region where the cost value is not guaranteed, and in that zone $E[y_f^2]$ is above the cutting value 1.42 shown in the figure.

Injecting a disturbance in the nominal and in the perturbed systems with the compensator obtained for $k = 60$, the outputs shown in Figure 8(c)-(d) are obtained. The improved performance of the VRC approach is clear from the plots.

PROOFS

Proof of Theorem 1

To shorten the notation, let $\boldsymbol{\delta} := (\delta_1, \ldots, \delta_N)$. Note that θ_k^*, ℓ_k^* are stochastic elements depending on δ, although to simplify the notation such a dependence is not explicitly indicated.

Define $B = \{\boldsymbol{\delta} \in \Delta^N$: there exists $k \in \{0, 1, \ldots, \bar{k}\}$ such that $V(\theta_k^*, \ell_k^*) > \varepsilon_k\}$, that is, B is the set of multiextractions $\boldsymbol{\delta}$ from Δ^N leading for some k to a violation bigger than ε_k. In these notations, the theorem statement writes $\Pr^N\{B\} \le \beta$.

Let

$$B_k = \{\boldsymbol{\delta} \in \Delta^N : V(\theta_k^*, \ell_k^*) > \varepsilon_k\}$$

be the event where the cost violation probability for a given k is bigger than ε_k, then $B = \bigcup_{k=0}^{\bar{k}} B_k$, which leads to the bound

$$\Pr^N\{B\} \le \sum_{k=0}^{\bar{k}} \Pr^N\{B_k\}. \tag{24}$$

The theorem is proved by first computing $\Pr^N\{B_k\}$ for $k = 0, 1, \ldots, \bar{k}$, and then by summing over k. Precisely, the inequality

$$\Pr^N\{B_k\} \le \binom{d+k}{k} \sum_{i=0}^{d+k} \binom{N}{i} \varepsilon_k^i (1-\varepsilon_k)^{N-i} \tag{25}$$

is established, and then the result follows by substitution in (24), that is,

$$\begin{aligned} \Pr^N\{B\} &\le \sum_{k=0}^{\bar{k}} \left[\binom{d+k}{k} \sum_{i=0}^{d+k} \binom{N}{i} \varepsilon_k^i (1-\varepsilon_k)^{N-i} \right] \\ &= \sum_{k=0}^{\bar{k}} \frac{\beta}{\bar{k}+1} = \beta, \end{aligned}$$

where the last equation follows from (11). Thus, to complete the proof, we have to establish the fundamental relation (25).

Fix a value for k. Given a subset $I = \{i_1, \ldots, i_k\}$ of k indexes from $\{1, \ldots, N\}$ (if $k = 0$ let $I = \emptyset$, the empty set), denote by θ_I^* the solution to the min-max problem where the scenarios with index in I have been removed, that is,

$$\theta_I^* := \arg \min_{\theta \in \mathbb{R}^d} \max_{i \in \{1,\ldots,N\} - I} \ell(\theta, \delta_i), \tag{26}$$

and let ℓ_I^* be the corresponding cost value, that is,

$$\ell_I^* := \max_{i \in \{1,\ldots,N\} - I} \ell(\theta_I^*, \delta_i).$$

Moreover, let

$$\Delta_I^N = \{\boldsymbol{\delta} \in \Delta^N : \ell(\theta_I^*, \delta_i) > \ell_I^*, \text{ for all } i \in I\}.$$

Thus, a $\boldsymbol{\delta}$ is in Δ_I^N if the cost value ℓ_I^* is violated in correspondence of all the scenarios in I that have been removed in the construction (26) of θ_I^*. Since the pair θ_k^*, ℓ_k^* generated by the VRC algorithm are such that the cost value ℓ_k^* is violated in correspondence of exactly k scenarios, it is clear that $(\theta_k^*, \ell_k^*) = (\theta_I^*, \ell_I^*)$ for some I such that $\boldsymbol{\delta} \in \Delta_I^N$. Thus,

$$\begin{aligned} B_k &= \{\boldsymbol{\delta} \in \Delta^N : V(\theta_k^*, \ell_k^*) > \varepsilon_k\} \\ &\subseteq \bigcup_{I \in \mathcal{I}} \{\boldsymbol{\delta} \in \Delta_I^N : V(\theta_I^*, \ell_I^*) > \varepsilon_k\} \end{aligned} \tag{27}$$

up to a zero probability set, where $\mathcal{I}$ is the collection of all possible choices of k indexes from $\{1, \ldots, N\}$.

The bound that is sought for $\Pr^N\{B_k\}$ can now be obtained by first bounding $\Pr^N\{\boldsymbol{\delta} \in \Delta_I^N : V(\theta_I^*, \ell_I^*) > \varepsilon_k\}$, and then summing over $I \in \mathcal{I}$.

Fix an $I = \{i_1, \ldots, i_k\}$ and write

$$\begin{aligned} &\Pr^N\{\boldsymbol{\delta} \in \Delta_I^N : V(\theta_I^*, \ell_I^*) > \varepsilon_k\} \\ &\quad = \int_{(\varepsilon_k, 1]} \Pr^N\{\Delta_I^N \,|\, V(\theta_I^*, \ell_I^*) = v\} dF_V(v) \\ &\quad = \int_{(\varepsilon_k, 1]} \Pr^N\{\ell(\theta_I^*, \delta_i) > \ell_I^*, \\ &\qquad \text{for all } i \in I \,|\, V(\theta_I^*, \ell_I^*) = v\} dF_V(v), \end{aligned} \tag{28}$$

where F_V is the cumulative distribution function of the random variable $V(\theta_I^*, \ell_I^*)$, and $\Pr^N\{\Delta_I^N \,|\, V(\theta_I^*, \ell_I^*) = v\}$ is the conditional probability of the event Δ_I^N under the condition that $V(\theta_I^*, \ell_I^*) = v$ (see [68, chap. II. sec. 7, (17)]). To evaluate the integrand in (28), remember that $V(\theta_I^*, \ell_I^*) = v$ means that $\Pr\{\delta : \ell(\theta_I^*, \delta) > \ell_I^*\} = v$; then, owing to the independence of the scenarios, the integrand equals v^k. Substituting in (28) yields

$$\Pr^N\{\boldsymbol{\delta} \in \Delta_I^N : V(\theta_I^*, \ell_I^*) > \varepsilon_k\} = \int_{(\varepsilon_k,1]} v^k dF_V(v). \quad (29)$$

To proceed, appeal to a result on F_V from [59], namely,

$$F_V(v) \geq \overline{F}_V(v) := 1 - \sum_{i=0}^{d} \binom{N-k}{i} v^i (1-v)^{N-k-i}. \quad (30)$$

This result follows from Theorem 1 in [59] by noting that the min-max problem $\min_{\theta \in \mathbb{R}^d} \max_{i \in \{1,\ldots,N\}-I} \ell(\theta, \delta_i)$ can be rewritten as

$$\begin{aligned} &\min_{\theta \in \mathbb{R}^d, h \in \mathbb{R}} h \\ &\text{subject to: } \ell(\theta, \delta_i) \leq h,\ i \in \{1,\ldots,N\} - I, \end{aligned}$$

that is, an optimization problem with $d+1$ variables and $N-k$ constraints. In addition, in [59] it is shown that the inequality in (30) is tight, that is, (30) holds with equality for a whole class of problems that are called fully supported in [59, Def. 3]. Now, the integrand v^k in (29) is an increasing function of v, so that $F_V(v) \geq \overline{F}_V(v)$ implies that

$$\int_{(\varepsilon_k,1]} v^k dF_V(v) \leq \int_{(\varepsilon_k,1]} v^k d\overline{F}_V(v). \quad (31)$$

This inequality can be verified by the following calculation, where the first equation follows from [68, Chap. ll, Sec. 6, Thrm. 11],

$$\begin{aligned} \int_{(\varepsilon_k,1]} v^k dF_V(v) &= 1 - \varepsilon_k^k F_V(\varepsilon_k) - \int_{(\varepsilon_k,1]} F_V(v) k v^{k-1} dv \\ &\leq 1 - \varepsilon_k^k \overline{F}_V(\varepsilon_k) - \int_{(\varepsilon_k,1]} \overline{F}_V(v) k v^{k-1} dv \\ &= \int_{(\varepsilon_k,1]} v^k d\overline{F}_V(v). \end{aligned}$$

Now, using (29) and (31),

$$\Pr^N\{\boldsymbol{\delta} \in \Delta_I^N : V(\theta_I^*, \ell_I^*) > \varepsilon_k\} \leq \int_{(\varepsilon_k,1]} v^k d\overline{F}_V(v).$$

Since the density of $\overline{F}_V$ is $(N-k-d)\binom{N-k}{d} v^d (1-v)^{N-k-d-1}$, this last integral is computed as

$$\int_{(\varepsilon_k,1]} (N-k-d)\binom{N-k}{d} v^{k+d} (1-v)^{N-k-d-1} dv,$$

which, by integration by parts, finally gives

$$\begin{aligned} &\Pr^N\{\boldsymbol{\delta} \in \Delta_I^N : V(\theta_I^*, \ell_I^*) > \varepsilon_k\} \\ &\quad \leq \frac{\binom{N-k}{d}}{\binom{N}{d+k}} \sum_{i=0}^{d+k} \binom{N}{i} \varepsilon_k^i (1-\varepsilon_k)^{N-i}. \end{aligned} \quad (32)$$

To conclude the proof, observe that $\mathcal{I}$ in (27) contains $\binom{N}{k}$ elements, and thus, using (32),

$$\begin{aligned} \Pr^N\{B_k\} &\leq \sum_{I \in \mathcal{I}} \Pr^N\{\boldsymbol{\delta} \in \Delta_I^N : V(\theta_I^*, \ell_I^*) > \varepsilon_k\} \\ &\leq \binom{N}{k} \frac{\binom{N-k}{d}}{\binom{N}{d+k}} \sum_{i=0}^{d+k} \binom{N}{i} \varepsilon_k^i (1-\varepsilon_k)^{N-i} \\ &= \binom{d+k}{k} \sum_{i=0}^{d+k} \binom{N}{i} \varepsilon_k^i (1-\varepsilon_k)^{N-i}, \end{aligned}$$

which is (25).

Proof of (12)

The value

$$M := \left\lfloor \frac{2}{\bar{\varepsilon}} \left(d + \ln \frac{1}{\beta} + 1 \right) + \frac{4}{\bar{\varepsilon}} \ln \left(\frac{2}{\bar{\varepsilon}} \left(d + \ln \frac{1}{\beta} + 1 \right) \right) \right\rfloor + 1,$$

satisfies

$$M \geq \frac{2\mu}{\bar{\varepsilon}} + \frac{2}{\bar{\varepsilon}} \cdot 2 \cdot \ln \left(\frac{2\mu}{\bar{\varepsilon}} \right),$$

where $\mu = d + \ln(1/\beta) + 1$. The inequality $2 \geq \mu/(\mu - 1)$ implies that

$$\begin{aligned} M &\geq \frac{2\mu}{\bar{\varepsilon}} + \frac{2}{\bar{\varepsilon}} \cdot \frac{\mu}{\mu - 1} \ln \left(\frac{2\mu}{\bar{\varepsilon}} \right) \\ &\geq \frac{2}{\bar{\varepsilon}} \cdot \frac{\mu}{\mu - 1} \left[\mu - 1 + \ln \left(\frac{2\mu}{\bar{\varepsilon}} \right) \right] \\ &= \frac{1}{\frac{\bar{\varepsilon}}{2} - \frac{1}{\frac{2\mu}{\bar{\varepsilon}}}} \left[\mu - 1 + \ln \left(\frac{2\mu}{\bar{\varepsilon}} \right) \right], \end{aligned}$$

that is,

$$M \frac{\bar{\varepsilon}}{2} \geq \mu - 1 + \ln \left(\frac{2\mu}{\bar{\varepsilon}} \right) + \frac{1}{\frac{2\mu}{\bar{\varepsilon}}} M.$$

Since $1 - (2\mu/\bar{\varepsilon}) \leq 0$, and observing that $\ln(x) + (1/x)(y - x) \geq \ln(y)$,

$$\begin{aligned} M \frac{\bar{\varepsilon}}{2} &\geq \mu - 1 + \ln \left(\frac{2\mu}{\bar{\varepsilon}} \right) + \frac{1}{\frac{2\mu}{\bar{\varepsilon}}} \left(M + 1 - \frac{2\mu}{\bar{\varepsilon}} \right) \\ &\geq \mu - 1 + \ln(M+1) \\ &= d + \ln \frac{1}{\beta} + \ln(M+1). \end{aligned}$$

Hence,

$$M \frac{\bar{\varepsilon}}{2} - d \geq \ln \frac{M+1}{\beta},$$

which, in view of

$$\frac{(M\bar{\varepsilon} - d)^2}{2M\bar{\varepsilon}} \geq M \frac{\bar{\varepsilon}}{2} - d,$$

implies that

$$\frac{(M\bar{\varepsilon} - d)^2}{2M\bar{\varepsilon}} \geq \ln \frac{M+1}{\beta}.$$

Taking the exponential of both sides gives

$$e^{-\frac{(M\bar{\varepsilon}-d)^2}{2M\bar{\varepsilon}}} \le \frac{\beta}{M+1},$$

which, by applying the Chernoff's bound for the binomial tail [69], [70] gives

$$\sum_{i=0}^{d}\binom{M}{i}\bar{\varepsilon}^i(1-\bar{\varepsilon})^{M-i} \le \frac{\beta}{M+1} \le \frac{\beta}{\alpha\%M+1}.$$

Thus, M satisfies (10); since the N selected in point 1 of the VRC algorithm is the smallest integer satisfying (10), we have $N \le M$.

Proof of (13)

If $\varepsilon_k < (k+d)/N$, then (13) is trivially true. If instead $\varepsilon_k \ge (k+d)/N$, then the Chernoff bound for the binomial tail [69], [70] applies, which gives

$$\sum_{i=0}^{d+k}\binom{N}{i}\varepsilon_k^i(1-\varepsilon_k)^{N-i} \le e^{-\frac{(N\varepsilon_k-d-k)^2}{2N\varepsilon_k}}.$$

Moreover, it holds that

$$\binom{d+k}{k} \le \frac{(d+k)^d e^d}{d^d},$$

so that the left-hand side of (11) is bounded by

$$\frac{(d+k)^d e^d}{d^d} e^{-\frac{(N\varepsilon_k-d-k)^2}{2N\varepsilon_k}}.$$

Hence,

$$\frac{(d+k)^d e^d}{d^d} \cdot e^{-\frac{(N\varepsilon_k-d-k)^2}{2N\varepsilon_k}} \ge \frac{\beta}{\overline{k}+1}.$$

This inequality can be rewritten as

$$(N\varepsilon_k-d-k)^2 \le 2N\varepsilon_k \ln\left[\frac{\overline{k}+1}{\beta}\cdot\frac{(d+k)^d e^d}{d^d}\right],$$

which, when written explicitly for ε_k, gives

$$\varepsilon_k \le \frac{k}{N} + \frac{d+h+\sqrt{h^2+2(d+k)h}}{N},$$

where we have defined

$$h = \ln\left[\frac{\overline{k}+1}{\beta}\cdot\frac{(d+k)^d e^d}{d^d}\right] = \ln(\overline{k}+1) + \ln\frac{1}{\beta} + d\left[1+\ln\frac{d+k}{d}\right].$$

ACKNOWLEDGMENTS

This work was supported by Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR), under project "Nuovi algoritmi ed applicazioni di identificazione e controllo adattativo" and by the European Union under project FP7 257005 "MoVeS: Modeling, Verification, and Control of Complex Systems."

AUTHOR INFORMATION

Simone Garatti is an assistant professor at the Dipartimento di Elettronica ed Informazione of the Politecnico di Milano. He received the Laurea degree and the Ph.D. degree in information technology engineering in 2000 and 2004, respectively, both from the Politecnico di Milano, Italy. He has been a visiting scholar at Lund University of Technology, Sweden, University of California, San Diego, Massachusetts Institute of Technology, Cambridge, and Northeastern University, Boston. He is a member of the IEEE Technical Committee on Computational Aspects of Control System Design and of the IFAC Technical Committee on Modeling, Identification, and Signal Processing. His research interests include system identification and model quality assessment, identification of interval predictor models, and randomized optimization for problems in systems and control.

Marco C. Campi (marco.campi@ing.unibs.it) is a professor of automatic control at the University of Brescia, Italy. In 1988, he received the doctor degree in electronic engineering from the Politecnico di Milano, Italy. From 1989 to 1992, he was a research fellow at the Centro di Teoria dei Sistemi of the National Research Council in Milano, and in 1992 he joined the University of Brescia, Italy. He is the chair of the IFAC Technical Committee on Modeling, Identification, and Signal Processing, has served as a Distinguished Lecturer of the IEEE Control Systems Society, and has delivered plenary and semiplenary addresses on topics related to randomization at major conferences including SYSID, MTNS, and CDC. In 2008, he received the IEEE CSS George S. Axelby Outstanding Paper Award for the article "The Scenario Approach to Robust Control Design." He is a Fellow of IEEE, a member of IFAC, and a member of SIDRA. His research interests include randomized methods, robust convex optimization, system identification, robust control and estimation, and learning theory. He can be contacted at the Dipartimento di Ingegneria dell'Informazione, Università di Brescia, via Branze 38, 25123 Brescia, Italy.

REFERENCES

[1] G. Zames, "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses," *IEEE Trans. Autom. Contr.*, vol. 26, no. 2, pp. 301–320, 1981.

[2] J. C. Doyle, "Guaranteed margins for LQG regulators," *IEEE Trans. Autom. Contr.*, vol. 23, no. 4, pp. 756–757, 1978.

[3] M. Green and D. J. N. Limebeer, *Linear Robust Control*. Upper Saddle River, NJ: Prentice-Hall, 1995.

[4] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[5] P. Colaneri, J. C. Geromel, and A. Locatelli, *Control Theory and Design: An RH_2 and RH_∞ Viewpoint*. San Diego, CA: Academic, 1997.

[6] D. Bertsimas and D. B. Brown, "Constructing uncertainty sets for robust linear optimization," *Oper. Res.*, vol. 57, no. 6, pp. 1483–1495, 2009.

[7] W. Chen, M. Sim, J. Sun, and C. P. Teo, "From CVAR to uncertainty sets: Implications in joint chance-constrained optimization," *Oper. Res.*, vol. 58, no. 2, pp. 470–485, 2010.

[8] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization, " *SIAM Rev.*, vol. 53, no. 3, pp. 464–501, 2011.

[9] K. J. Åström, *Introduction to Stochastic Control Theory*. London, U.K.: Academic, 1970.
[10] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification and Adaptive Control*. Upper Saddle River, NJ: Prentice-Hall, 1986.
[11] D. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2005.
[12] M. Vidyasagar, "Statistical learning theory and randomized algorithms for control," *IEEE Control Syst. Mag.*, vol. 18, no. 6, pp. 69–85, 1998.
[13] M. Vidyasagar, "Randomized algorithms for robust controller synthesis using statistical learning theory," *Automatica*, vol. 37, no. 10, pp. 1515–1528, 2001.
[14] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control* (Lecture Notes in Control and Information Science), vol. 245, A. Garulli, A. Tesi, and A. Vicino, Eds. London, U.K.: Springer-Verlag, 1999, pp. 207–226.
[15] P. O. M. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Autom. Contr.*, vol. 43, no. 8, pp. 1136–1142, 1998.
[16] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.
[17] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, no. 4, pp. 523–533, 2006.
[18] D. Bertsimas and D. B. Brown, "Constrained stochastic LQC: A tractable approach," *IEEE Trans. Autom. Contr.*, vol. 52, no. 10, pp. 1826–1841, 2007.
[19] S. I. Marcus, E. Fernández-Gaucherand, D. Hernández-Hernández, S. Coraluppi, and P. Fard, "Risk sensitive Markov decision processes," in *Systems and Control in the Twenty-First Century*, C. I. Byrnes, B. N. Datta, and D. S. Gillian, eds., vol. 22. Boston, MA: Birkhäuser, 1997, pp. 263–279.
[20] D. H. Jacobson, "Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games," *IEEE Trans. Autom. Contr.*, vol. 18, no. 2, pp. 124–131, 1973.
[21] P. Whittle, "Risk-sensitive linear/quadratic/Gaussian control," *Adv. Appl. Probab.*, vol. 13, no. 4, pp. 764–777, 1981.
[22] M. R. James, "Asymptotic analysis of nonlinear stochastic risk sensitive control and differential games," *Math. Control, Signals, Syst.*, vol. 5, no. 4, pp. 401–417, 1992.
[23] M. C. Campi and M. R. James, "Nonlinear discrete-time risk-sensitive optimal control," *Int. J. Robust Nonlinear Control*, vol. 6, no. 1, pp. 1–19, 1996.
[24] R. F. Stengel, "Some effects of parameter variations on the lateral-directional stability of aircraft," *AIAA J. Guid. Control*, vol. 3, no. 2, pp. 124–131, 1980.
[25] R. F. Stengel and L. R. Ray, "Stochastic robustness of linear time-invariant control systems," *IEEE Trans. Autom. Contr.*, vol. 36, no. 1, pp. 82–87, 1991.
[26] L. R. Ray and R. F. Stengel, "A Monte Carlo approach to the analysis of control system robustness," *Automatica*, vol. 29, no. 1, pp. 229–236, 1993.
[27] B. R. Barmish and C. M. Lagoa, "The uniform distribution: A rigorous justification for its use in robustness analysis," *Math. Control, Signals, Syst.*, vol. 10, no. 3, pp. 203–222, 1997.
[28] C. M. Lagoa, P. S. Shcherbakov, and B. R. Barmish, "Probabilistic enhancement of classical robustness margins: The unirectangularity concept," *Syst. Control Lett.*, vol. 35, no. 1, pp. 31–43, 1998.
[29] P. P. Khargonekar and A. Tikku, "Randomized algorithms for robust control analysis have polynomial time complexity," in *Proc 35th IEEE Conf. Decision Control*, 1996, pp. 3470–3475.
[30] R. Tempo, E. W. Bai, and F. Dabbene, "Probabilistic robustness analysis: Explicit bounds for the minimum number of samples," *Syst. Control Lett.*, vol. 30, no. 5, pp. 237–242, 1997.
[31] G. Calafiore and B. T. Polyak, "Stochastic algorithms for exact and approximate feasibility of robust LMIs," *IEEE Trans. Autom. Contr.*, vol. 46, no. 11, pp. 1755–1759, 2001.
[32] B. T. Polyak and R. Tempo, "Probabilistic robust design with linear quadratic regulators," *Syst. Control Lett.*, vol. 43, no. 5, pp. 343–353, 2001.
[33] Y. Fujisaki, F. Dabbene, and R. Tempo, "Probabilistic robust design of LPV control systems," *Automatica*, vol. 39, no. 8, pp. 1323–1337, 2003.
[34] S. Kanev, B. De Schutter, and M. Verhaegen, "An ellipsoid algorithm for probabilistic robust controller design," *Syst. Control Lett.*, vol. 49, no. 5, pp. 365–375, 2003.
[35] Y. Fujisaki and Y. Oishi, "Guaranteed cost regulator design: A probabilistic solution and a randomized algorithm," *Automatica*, vol. 43, no. 2, pp. 317–324, 2007.
[36] Y. Oishi, "Polynomial-time algorithms for probabilistic solutions of parameter-dependent linear matrix inequalities," *Automatica*, vol. 43, no. 3, pp. 538–545, 2007.
[37] C. M. Lagoa and B. R. Barmish, "Distributionally robust Monte Carlo simulation: A tutorial survey," in *Proc. 15th IFAC World Congr.*, 2002, pp. 151–162.
[38] A. Charnes and W. W. Cooper, "Chance constrained programming," *Manage. Sci.*, vol. 6, no. 1, pp. 73–79, 1959.
[39] A. Prèkopa, *Stochastic Programming*. Boston, MA: Kluwer, 1995.
[40] A. Prèkopa, "The use of discrete moment bounds in probabilistic constrained stochastic programming models," *Ann. Oper. Res.*, vol. 85, pp. 21–38, 1999.
[41] A. Prèkopa, "Probabilistic programming," in *Handbooks in Operations Research and Management Science*, vol. 10, A. Ruszczyǹski and A. Shapiro, Eds. London, U.K.: Elsevier, 2003, pp. 267–352.
[42] D. Dentcheva, "Optimization models with probabilistic constraints," in *Probabilistic and Randomized Methods for Design under Uncertainty*, G. Calafiore and F. Dabbene, Eds. London, U.K.: Springer-Verlag, 2006, pp. 47–96.
[43] G. Calafiore and M. C. Campi, "Uncertain convex programs: Randomized solutions and confidence levels," *Math. Program.*, vol. 102, no. 1, pp. 25–46, 2005.
[44] A. Shapiro, D. Dentcheva, and A. Ruszczyǹski, *Lectures on Stochastic Programming: Modelling and Theory*. Philadelphia, PA: SIAM, 2009.
[45] R. F. Stengel, L. R. Ray, and C. I. Marrison, "Probabilistic evaluation of control system robustness," *Int. J. Syst. Sci.*, vol. 26, no. 7, pp. 1363–1382, 1995.
[46] C. I. Marrison and R. F. Stengel, "Robust control system design using random search and genetic algorithms," *IEEE Trans. Autom. Contr.*, vol. 42, no. 6, pp. 835–839, 1997.
[47] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *Amer. Inst. Chem. Engineers J.*, vol. 45, no. 8, pp. 1743–1752, 1999.
[48] P. Li, M. Wendt, and G. Wozny, "A probabilistically constrained model predictive controller," *Automatica*, vol. 38, no. 7, pp. 1171–1176, 2002.
[49] T. Motoda, Y. Miyazawa, and R. F. Stengel, "Robust control system design using simulated annealing," *J. Guid. Control, Dyn.*, vol. 25, no. 2, pp. 267–274, 2002.
[50] Q. Wang and R. F. Stengel, "Robust control of nonlinear systems with parametric uncertainty," *Automatica*, vol. 38, no. 9, pp. 1591–1599, 2002.
[51] Q. Wang and R. F. Stengel, "Probabilistic control of nonlinear uncertain systems," in *Probabilistic and Randomized Methods for Design Under Uncertainty*, G. Calafiore and F. Dabbene, Eds. London, U.K: Springer-Verlag, 2006, pp. 381–414.
[52] L. Blackmore, H. Li, and B. C. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *Proc. American Control Conf.*, Minneapolis, MN, 2006, pp. 2831–2837.
[53] M. C. Campi and S. Garatti, "Modulating robustness in robust control: Making it easy through randomization," in *Proc. 46th IEEE Conf. Decision Control*, 2007, pp. 2164–2168.
[54] T. Alamo, R. Tempo, and E. F. Camacho, "Revisiting statistical learning theory for uncertain feasibility and optimization problems," in *Proc. 46th IEEE Conf. Decision Control*, 2007, pp. 2135–2140.
[55] T. Alamo, R. Tempo, and E. F. Camacho, "Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems," *IEEE Trans. Autom. Contr.*, vol. 54, no. 11, pp. 2545–2559, 2009.
[56] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance constrained stochastic predictive control," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 502–517, 2010.
[57] M. C. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality," *J. Optim. Theory Appl.*, vol. 148, no. 2, pp. 257–280, 2011.
[58] G. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Trans. Autom. Contr.*, vol. 51, no. 5, pp. 742–753, 2006.
[59] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1211–1230, 2008.
[60] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design, *Annu. Reviews Control*, vol. 33, pp. 149–157, 2009,.
[61] G. Calafiore, F. Dabbene, and R. Tempo, "Randomized algorithms for probabilistic robustness with real and complex structured uncertainty," *IEEE Trans. Autom. Contr.*, vol. 45, no. 12, pp. 2218–2235, 2000.
[62] G. Calafiore and F. Dabbene, "A probabilistic framework for problems with real structured uncertainty in systems and control," *Automatica*, vol. 38, no. 8, pp. 1265–1276, 2002.

[63] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*. London, U.K.: Springer-Verlag, 2005.
[64] M. C. Campi, "Why is resorting to fate wise? A critical look at randomized algorithms in systems and control," *Eur. J. Control*, vol. 16, no. 5, pp. 419–430, 2010.
[65] M. Grant and S. Boyd. (2010, Apr.). CVX: Matlab software for disciplined convex programming, version 1.21. [Online]. Available: http://cvxr.com/cvx
[66] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control* (Lecture Notes in Control and Information Sciences), vol. 371, V. Blondel, S. Boyd, and H. Kimura, Eds. London, U.K.: Springer-Verlag, 2008, pp. 95–110.
[67] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. CACSD Conf.*, 2004, pp. 284–289.
[68] A. N. Shiryaev, *Probability*. New York: Springer-Verlag, 1996.
[69] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Ann. Mathematical Statistics*, vol. 23, pp. 493–507, 1952.
[70] M. Vidyasagar, *A Theory of Learning and Generalization*. London, U.K.: Springer-Verlag, 1997.

Callouts:

The cost-violation plot provides a quantitative support to guide the user to make this decision.

Averaging is not the only use one can make of probability, and an alternative approach consists of using probability to quantify the chance that a certain performance specification is attained.

The level of robustness is adjustable, and so this approach is called a modulating robustness approach.

The VRC algorithm has determined a region of small volume whose elimination guarantees a large improvement in the optimal value.