Available online at www.sciencedirect.com
ScienceDirect
Journal of Complexity 23 (2007) 301–316
Journal of COMPLEXITY
www.elsevier.com/locate/jco

A survey of randomized algorithms for control synthesis and performance verification☆

Giuseppe Calafiore[a], Fabrizio Dabbene[b], Roberto Tempo[b,*]

[a] *Dipartimento di Automatica e Informatica, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy*
[b] *IEIIT-CNR, Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy*

Received 7 November 2006; accepted 8 January 2007
This paper is dedicated to Professor Henryk Woźniakowski on the occasion of his 60th birthday
Available online 30 January 2007

Abstract

In this paper, we present an overview of probabilistic techniques based on randomized algorithms for solving "hard" problems arising in performance verification and control of complex systems. This area is fairly recent, even though its roots lie in the robustness techniques for handling uncertain control systems developed in the 1980s. In contrast to these deterministic techniques, the main ingredient of the methods discussed in this survey is the use of probabilistic concepts. The introduction of probability and random sampling permits overcoming the fundamental tradeoff between numerical complexity and conservatism that lie at the roots of the worst-case deterministic methodology. The simplicity of implementation of randomized techniques may also help bridging the gap between theory and practical applications.

Keywords: Complex control systems; Randomized algorithms

1. Preliminaries and motivations

Information-based complexity (IBC) is a general theory, developed within computer science, that studies the complexity of approximately solving problems in the presence of partial and/or contaminated information, see [54,55] and references therein. Typical applications of IBC are

☆ Preliminary versions of this paper have been published in the Proceedings of the IEEE Conference of Decision and Control, Maui, 2003 and of the Workshop on New Approaches and Problems in System and Control Theory, Istanbul, 2005.

* Corresponding author. Fax: +39 011 564 5429.
E-mail addresses: giuseppe.calafiore@polito.it (G. Calafiore), fabrizio.dabbene@polito.it (F. Dabbene), tempo@polito.it (R. Tempo).

0885-064X/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jco.2007.01.003

distributed computation, clock synchronization, solution of nonlinear equations and integration problems. Other applications of IBC include system identification, time series analysis and control of uncertain systems. Earlier contributions developing optimal algorithms for control-oriented robust identification in the presence of noisy data within the general setting of IBC are [36,49,24]. While [36,49] studied optimal algorithms for parametric identification and filtering in the presence of noisy data, [24] focused on nonparametric H_∞ identification. These papers, however, followed a deterministic approach. Other noticeable contributions not directly related to IBC but still linking optimality and complexity include [64].

IBC originally focused on the worst-case setting, see [56], but later expanded to other settings: average-case, probabilistic and asymptotic. In each setting, the task is to develop optimal algorithms and to compute the associated errors and computational complexity. It is interesting to observe that a similar path has been followed within the area of robust control. The first main contributions dealt with a worst-case approach, see [65] for the pioneering work of H_∞ control, but more recently several authors have discussed different settings for handling complex uncertain system. In particular, in this paper we survey the probabilistic, or randomized, approach to control design, see [52].

Many pessimistic results on the complexity-theoretic barriers to (deterministic) robustness problems [6,37,43] have stimulated research in the direction of finding "ways around'' these limits; see also [57] for discussions regarding the "curse of dimensionality'' within computer science and related areas. One option is, for instance, to first shift the meaning of robustness from its usual deterministic sense to a probabilistic one. In this respect, we would claim that a certain property of a control system is "almost'' robustly satisfied, if it holds for "most'' of the instances of the uncertainty. In other words, we accept the risk of this property being violated by a set of uncertainties having small probability measure. Such systems may be viewed as being *practically robust* from an engineering point of view. This shift in meaning implies a statistical description of the uncertainty, which is deemed more natural and engineeringly sound than a purely unknown-but-bounded one. Hence, one of the advantages of the probabilistic approach is to provide a rethinking of the relation between the stochastic and the robust paradigms, utilizing classical worst-case bounds of robust control together with probabilistic information, which is often neglected in the deterministic context. The interplay of probability and robustness also leads to innovative concepts such as the probabilistic robustness margin and the probability degradation function, see, for instance, [9,14].

However, it should be noticed that moving from deterministic to probabilistic robustness does *not* automatically imply a simplification of the problem. Indeed, assessing probabilistic robustness of a given property may be even computationally harder than establishing robustness in the usual deterministic sense, since it requires the exact computation of a multi-dimensional probability integral [17]. It is at this point that *randomization* comes into play: the performance probability is *estimated* by randomly sampling the uncertainties, and tail inequalities are used to bound the estimation error. Since the estimated probability is itself a random quantity, this method always entails a certain *risk of failure*, i.e. there exists a nonzero probability of making an erroneous estimation. These algorithms, usually called *randomized algorithms* (RAs), often have low complexity and are associated to robustness bounds which are less conservative than the classical ones, obviously at the expense of a probabilistic risk.

Randomization can also be effectively used for control *synthesis*. Basically, three different methodologies are being developed for this purpose: the approach based on the Vapnik–Chervonenkis theory of learning [30,60–63], the scenario approach of [8], and the sequential approximation methods based on gradient iterations [15,21,22,26,27,33,40,45,58] or ellipsoid

iterations [28]. These latter algorithms fall in the general family of stochastic optimization programs [31].

From the historical point of view, probabilistic methods for robustness made some early appearance in the eighties, but at that time they did not receive adequate attention in the systems and control literature. In particular, the notion of "probability of instability,'' which is crucial for stochastic robustness, was first introduced in the context of flight control in 1980 by Stengel [46] and then revisited in his book on stochastic optimal control in 1986 [47]. In 1989, the paper [20] by Djavdan et al. titled "Probabilistic robust controller design'' was published. This is presumably the first paper with a title containing both words "probabilistic'' and "robust.'' Subsequently, Stengel and coauthors continued the line of research on stochastic robustness exploring various techniques, mainly based on Monte Carlo, for the computation of the probability of instability and with specific attention to flight dynamics applications in aerospace engineering. However, the absence of newly developed mathematical tools limited these attempts to merge probability and robustness to *analysis* problems.

A few years later, in 1996 the papers [29,50] (developed independently by Khargonekar and Tikku and by Tempo et al.) proposed an approach based on explicit sample size bounds, thus refuelling enthusiasm on randomized techniques. Subsequently, the study of statistical learning theory and its application to robust control conducted by Vidyasagar [60–62] provided additional impetus and also exposed researchers of robust control to a different viewpoint based on solid mathematical foundations. This formulation led to the development of RAs for control system *design*.

RAs and probabilistic methods are now becoming an increasingly popular tool within the robust control community. For a comprehensive treatment of these topics the reader is referred to the recently published books [12,52]. A different perspective is given in [53] where the class of *Las Vegas RAs* is introduced and studied. Interesting results regarding RAs for system identification and model validation are given in [7,48]. Mixed methods, which successfully combine deterministic and probabilistic techniques, are analyzed in [18,23] for solving the so-called one-in-a-box problem and for computing a fixed order stabilizer. Finally, specific applications of RAs for control systems include unmanned aerial vehicles (UAV) platform stabilization [34], robustness of high-speed networks [2] and synthesis of real-time embedded controllers [42].

2. RAs for robustness analysis

In our exposition, we denote by RA any algorithm that makes some random choices in the course of its execution. The outcome of such a decision process is hence a random variable, and the quality of this outcome is to be assessed via a probabilistic statement.

Specifically, an RA for probabilistic robustness *analysis* is an algorithm that, based on random extractions of uncertainty samples, returns an estimated probability of satisfaction of an analysis condition. The estimate provided by the RA is within an a priori specified *accuracy* $\varepsilon \in (0, 1)$ from the true value, with high *confidence* $1 - \delta$, $\delta \in (0, 1)$. That is, the algorithm may indeed fail to return an approximately correct estimate, but with probability at most δ.

Formally, let $\mathbf{\Delta} \in \mathbb{D}$ represent the random uncertainty acting on the system, where $\mathbb{D}$ is the support of the random variable $\mathbf{\Delta}$ (for instance, $\mathbb{D}$ can be the space of ℓ-dimensional real vectors, or the space of block-structured matrices with a norm bound), and denote with $f_{\mathbf{\Delta}}(\Delta)$ the probability density function (pdf) of $\mathbf{\Delta}$. Let further $J(\Delta) : \mathbb{D} \to \mathbb{R}$ be a *performance function* for the uncertain system, i.e. a function measuring the performance of the system for a given value of Δ. For instance, $J(\Delta)$ can be the $\mathcal{H}_2$ or the $\mathcal{H}_\infty$ norm of the system.

First, we specify the characteristics that an RA for probabilistic performance verification should comply with.

Definition 1 (*RA for probabilistic performance verification*). Let $\varepsilon \in (0, 1)$, $\delta \in (0, 1)$ be probability levels. Given a performance level γ, the RA should return with probability $1-\delta$ an estimate $\hat{\mathbf{p}}_N$ of the probability of performance

$$p = \Pr\{J(\mathbf{\Delta}) \leqslant \gamma\}$$

that is within ε accuracy from p. The estimate $\hat{\mathbf{p}}_N$ should be constructed based on a finite number N of random samples of $\mathbf{\Delta}$.

Notice that a simple RA for performance verification is directly constructed by means of the classical Monte Carlo method as follows.

Algorithm 1. *Given ε, δ, γ, the following RA returns with probability $1-\delta$ an estimate $\hat{\mathbf{p}}_N$ such that*

$$|p - \hat{\mathbf{p}}_N| \leqslant \varepsilon \tag{1}$$

1. *Determine a suitable sample size $\overline{N} = \overline{N}_{\mathrm{ch}}(\varepsilon, \delta)$ according to* (3);
2. *Draw $N \geqslant \overline{N}$ independent samples $\mathbf{\Delta}^{(1)}, \ldots, \mathbf{\Delta}^{(N)}$ distributed according to $f_{\mathbf{\Delta}}(\Delta)$;*
3. *Return the empirical probability*

$$\hat{\mathbf{p}}_N = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[J(\mathbf{\Delta}^{(i)}) \leqslant \gamma]$$

where $\mathbb{I}[\cdot] = 1$ when its argument is true, and it is zero otherwise.

It is shown in Section 2.1 that, when the sample size $\overline{N}_{\mathrm{ch}}(\varepsilon, \delta)$ is chosen according to (3), then Algorithm 1 indeed satisfies the requirements of Definition 1.

A related robustness analysis problem is to assess the worst-case performance level of the system. In this case, we shall consider an RA that enjoys the following features.

Definition 2 (*RA for probabilistic worst-case performance*). Let $p^* \in (0, 1)$, $\delta \in (0, 1)$ be assigned probability levels. The RA should return with probability $1-\delta$ a performance level $\hat{\gamma}_N$ such that

$$\Pr\left\{J(\mathbf{\Delta}) \leqslant \hat{\gamma}_N\right\} \geqslant p^*.$$

The performance level $\hat{\gamma}_N$ should be constructed based on a finite number N of random samples of $\mathbf{\Delta}$.

In words, we require that an RA for probabilistic worst-case performance determines a performance level $\hat{\gamma}_N$ which is guaranteed for most of the uncertainty instances. This can be obtained via the following algorithm.

Algorithm 2. *Given p^*, δ, the following RA returns with probability $1-\delta$ a level $\hat{\gamma}_N$ such that*

$$\Pr\left\{J(\mathbf{\Delta}) \leqslant \hat{\gamma}_N\right\} \geqslant p^* \tag{2}$$

1. *Determine a suitable sample size $\overline{N} = \overline{N}_{\rm wc}(p^*, \delta)$ according to* (4);
2. *Draw $N \geqslant \overline{N}$ independent samples $\mathbf{\Delta}^{(1)}, \ldots, \mathbf{\Delta}^{(N)}$ distributed according to $f_{\mathbf{\Delta}}(\Delta)$;*
3. *Return the empirical maximum*

$$\hat{\gamma}_N = \max_{i=1,\ldots,N} J(\mathbf{\Delta}^{(i)}).$$

It is shown in Section 2.1 that, when the sample size $\overline{N}_{\rm wc}(\varepsilon, \delta)$ is chosen according to (4), then Algorithm 1 indeed satisfies the requirements of Definition 2.

2.1. Computational complexity of RAs and related issues

The computational complexity of an RA is due to three main sources:

1. The minimum number of samples required to attain the desired probabilistic guarantees of quality in the solution (sample complexity).
2. The computational cost of generating the random samples of $\mathbf{\Delta}$ according to $f_{\mathbf{\Delta}}(\Delta)$.
3. The computational cost of *evaluating* the performance $J(\Delta)$, for fixed Δ.

The second of the above components depends critically on the type and structure of the set in which randomization is to be performed. This issue is discussed in Section 5. We remark that the choice of a uniform distribution is common in this setting, because it permits the interpretation of the probabilistic statements in terms of "volumes'' of certain sets, and also for its worst-case properties, see [4,5,32].

In all typical control applications, the performance function $J(\Delta)$ can be efficiently evaluated, for fixed Δ. Hence, we see that the actual computational burden of an RA is strictly related to the first item above, i.e. to its sample complexity. Assuming that the problem "size'' is described by some integer n (for instance, the size of a matrix or the order of a dynamical system), we say that the RA is *efficient* if its sample complexity is at most polynomial in the problem size n and the probabilistic levels.

For instance, for the two algorithms presented above, the sample complexity can be assessed using fairly classical large deviations inequalities. In particular, for Algorithm 1, one of the best-known bounds is the so-called Chernoff bound [16], derived from the Hoeffding tail probability inequality, see [25],

$$\overline{N}_{\rm ch}(\varepsilon, \delta) = \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}. \tag{3}$$

For Algorithm 2, the following bound was independently derived in [29,51]:

$$\overline{N}_{\rm wc}(p^*, \delta) = \frac{\ln \dfrac{1}{\delta}}{\ln \dfrac{1}{p^*}}. \tag{4}$$

An important conclusion is therefore that Algorithms 1 and 2 have a sample complexity that is not only polynomial in the problem size and probabilistic levels, but that is actually *independent* of the problem size (the dimension of $\mathbb{D}$).

3. RAs for control synthesis

Two different philosophies are currently followed for control synthesis in the probabilistic context. The first philosophy aims at designing controllers that satisfy the performance specification for most values of the uncertainties, i.e. that are robust in a probabilistic sense. In this case, an RA should return a controller, represented by a parameter $\theta \in \Theta$ ($\Theta \subseteq \mathbb{R}^{n_\theta}$ being the set of allowable design parameters), which guarantees the desired performance with an a priori specified (high) probability $p^* \in (0, 1)$. As in the analysis case, this algorithm may fail with probability at most δ.

An alternative synthesis paradigm is to seek a controller that (approximately) minimizes the mean value of the performance index. That is, the objective is to obtain a controlled system that performs well *on average*.

Let $J(\Delta, \theta) : \mathbb{D} \times \Theta \to \mathbb{R}$ be a function measuring the closed-loop performance of the system with controller parameter θ. An RA for the robust design approach is specified as follows.

Definition 3 (*RA for robust performance synthesis (RPS)*). Let γ be a given performance level and let $p^* \in (0, 1)$, $\delta \in (0, 1)$ be assigned probability levels. An RA for robust synthesis should return with probability $1 - \delta$ a design vector $\hat{\boldsymbol{\theta}}_N \in \Theta$ such that

$$\Pr\left\{J(\boldsymbol{\Delta}, \hat{\boldsymbol{\theta}}_N) \leqslant \gamma\right\} \geqslant p^*. \tag{5}$$

The controller parameter $\hat{\boldsymbol{\theta}}_N$ should depend on a finite number N of random samples of $\boldsymbol{\Delta}$.

An RA for the average performance synthesis (APS) should instead have the following characteristics.

Definition 4 (*RA for APS*). Let $\varepsilon \in (0, 1)$, $\delta \in (0, 1)$ be given probability levels. Let

$$\phi(\theta) \doteq E_{\boldsymbol{\Delta}}\left[J(\boldsymbol{\Delta}, \theta)\right]$$

denote the *average* performance (with respect to $\boldsymbol{\Delta}$) of the controlled plant, and

$$\phi^* \doteq \min_{\theta \in \Theta} \phi(\theta)$$

denote the optimal achievable average performance. A randomized average synthesis algorithm should return with probability $1 - \delta$ a design vector $\hat{\boldsymbol{\theta}}_N \in \Theta$ such that

$$\phi(\hat{\boldsymbol{\theta}}_N) - \phi^* \leqslant \varepsilon.$$

The controller parameter $\hat{\boldsymbol{\theta}}_N$ should be constructed based on a finite number N of random samples of $\boldsymbol{\Delta}$.

We present in the next sections some RAs that meet these specifications. We first deal with the average design approach.

3.1. The learning theory approach for APS

Consider a normalized performance function $J(\Delta, \theta) : \mathbb{D} \times \Theta \to [0, 1]$, which measures the performance of the controlled plant, for given Δ and θ. The goal is to determine a controller, i.e. a $\theta \in \Theta$, such that the *average* performance of the controlled plant is minimized. In the approach presented in this section, the design follows two steps: in the first step, the expected value $\phi(\theta) = E_{\mathbf{\Delta}}\left[J(\mathbf{\Delta}, \theta)\right]$ is estimated, and in the second step a minimization is performed on $\phi(\theta)$ to obtain the "optimal'' controller. The following RA for average synthesis satisfies the requirements of Definition 4.

Algorithm 3. *Given ε, δ, the following RA returns with probability $1 - \delta$ a controller parameter $\hat{\boldsymbol{\theta}}_N$ such that*

$$\phi(\hat{\boldsymbol{\theta}}_N) - \phi^* \leqslant \varepsilon \tag{6}$$

1. *Determine a suitable sample size $\overline{N} = \overline{N}_{\mathrm{sl}}(\varepsilon, \delta)$ according to* (8);
2. *Draw $N \geqslant \overline{N}$ independent samples $\mathbf{\Delta}^{(1)}, \ldots, \mathbf{\Delta}^{(N)}$ distributed according to $f_{\mathbf{\Delta}}(\Delta)$;*
3. *Return the controller parameter*

$$\hat{\boldsymbol{\theta}}_N = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} J(\mathbf{\Delta}^{(i)}, \theta). \tag{7}$$

The sample complexity of this algorithm may be assessed using the uniform convergence bounds derived in statistical learning theory [44,59,60,62], which yield

$$\overline{N}_{\mathrm{sl}}(\varepsilon, \delta) = \frac{128}{\varepsilon^2}\left[\ln\frac{8}{\delta} + d\left(\ln\frac{32\mathrm{e}}{\varepsilon} + \ln\ln\frac{32\mathrm{e}}{\varepsilon}\right)\right], \tag{8}$$

where d is an upper bound on the so-called Pollard (or P) dimension of the function family $\{J(\cdot, \theta), \theta \in \Theta\}$.

Notice that, in principle, the minimization over $\theta \in \Theta$ in (7) can be performed by any numerical optimization method. However, since this constrained minimization problem is in general non-convex, there are obvious difficulties in finding its global solution. Thus, a viable approach would be to use an RA also for this minimization. When even this latter optimization step is performed by means of randomization (over the controller parameters), then the global RA returns a so-called probably approximate near minimizer of the averaged performance function; see [30,60,61] for further discussions along this line.

Certainly, a notable feature of this fully randomized synthesis approach is its complete generality. However, two main criticisms are in order. The first one relates to the "looseness'' of

bound (8) which makes the required number of samples so large to be hardly useful in practice. Secondly, randomization over the controller parameters (which is in general necessary) leads to rather weak statements on the quality of the resulting solutions. This latter problem may be avoided in principle when the performance function $J(\Delta, \theta)$ is *convex* in θ for any fixed Δ.

In the next section, we discuss RAs for robust (as opposed to average) performance synthesis, which are based on sequential stochastic iterations. These algorithms are applicable whenever $J(\Delta, \theta)$ is *convex* in θ for any fixed Δ, and their convergence properties are *not* based on the Vapnik–Chervonenkis theory.

4. Sequential approaches for RPS

In this section, we outline the basic sequential stochastic algorithms that have been recently proposed in the literature for probabilistic robust synthesis. In particular, we recall the stochastic gradient methods introduced in [15,21,45], and the iterative ellipsoid method proposed in [28]. In this paper we show how these algorithms can be suitably modified in order to fit Definition 3.

Consider a performance function $J(\Delta, \theta) : \mathbb{D} \times \Theta \to \mathbb{R}$, *convex* in θ for all $\Delta \in \mathbb{D}$, and let γ be a desired performance level. The goal is here to develop RAs that conform to the requirements of Definition 3, i.e. that return a candidate controller $\hat{\boldsymbol{\theta}}_N \in \Theta$ such that the performance requirement $J(\boldsymbol{\Delta}, \hat{\boldsymbol{\theta}}_N) \leqslant \gamma$ is satisfied with probability at least p^*. The main idea behind the proposed design paradigm consists of randomization to handle uncertainty and convex optimization to select the design parameters. More precisely, we concentrate on design algorithms based on sequential random update techniques that aim to minimize, with respect to θ, a particular function $v(\Delta, \theta)$ related to $J(\Delta, \theta)$. This function, which measures the extent of violation of the performance specification $J(\Delta, \theta) \leqslant \gamma$, is called *performance violation function* and is now formally defined.

Definition 5 (*Performance violation function*). Given a performance function $J(\Delta, \theta)$ and a desired performance level γ, a function $v(\Delta, \theta) : \mathbb{D} \times \Theta \to \mathbb{R}$ is said to be a *performance violation function* if the following properties hold:

1. $v(\Delta, \theta) \geqslant 0$ for all $\Delta \in \mathbb{D}$ and $\theta \in \Theta$;
2. $v(\Delta, \theta) = 0$ if and only if $J(\Delta, \theta) \leqslant \gamma$.

Assume now that a robustly feasible solution exists, and in particular that there exists a ball of center $\theta^* \in \Theta$ and radius $r > 0$ contained in the solution set

$$\mathcal{S} = \{\theta \in \Theta : v(\Delta, \theta) = 0 \text{ for all } \Delta \in \mathbb{D}\}.$$

Notice that only a lower bound on the feasibility radius r is necessary for assessing the convergence of Algorithm 4 below. We further assume that for any $\theta \notin \mathcal{S}$, it holds that

$$\Pr_{\boldsymbol{\Delta}} \{v(\boldsymbol{\Delta}, \theta) > 0\} > 0.$$

That is, there is a strictly positive probability of detecting infeasibility of a candidate solution by random sampling the uncertainty $\boldsymbol{\Delta}$.

4.1. Stochastic gradient methods

Let $[\cdot]_\Theta$ denote the projection operator over the set Θ, and let $\partial^{(i,k)}$ denote the subgradient of $v(\Delta, \theta)$ with respect to θ, evaluated at $\mathbf{\Delta}^{(i)}$, $\boldsymbol{\theta}^{(k)}$, i.e.

$$\partial^{(i,k)} \doteq \partial_\theta \{v(\Delta, \theta)\}\big|_{\Delta=\mathbf{\Delta}^{(i)}, \theta=\boldsymbol{\theta}^{(k)}}.$$

The following algorithm returns a controller that fulfills the requirements of Definition 3.

Algorithm 4 (Stochastic gradient algorithm for RPS). *Given* $p^*, \delta \in (0, 1)$ *and* γ, *the following RA returns with probability at least* $1 - \delta$ *a design vector* $\hat{\boldsymbol{\theta}}_N$ *such that* (5) *holds.*

1. *Initialization.*
   - *Determine a sample size function* $N(k) = \overline{N}_{\rm ss}(p^*, \delta, k)$ *according to* (10);
   - *Set* $k = 0$, $i = 0$, *and choose* $\theta^{(0)} \in \Theta$;
2. *Feasibility loop.*
   - *Set* $\ell = 0$ *and* feas = true;
   - *While* $\ell < N(k)$ *and* feas = true
     - *Set* $i = i + 1$, $\ell = \ell + 1$;
     - *Draw* $\mathbf{\Delta}^{(i)}$ *distributed according to* $f_{\mathbf{\Delta}}$;
     - *If* $v(\mathbf{\Delta}^{(i)}, \boldsymbol{\theta}^{(k)}) > 0$ *set* feas = false;
   - *End While*;
3. *Exit condition.*
   - *If* feas = true
     - *Set* $N = i$;
     - *Return* $\hat{\boldsymbol{\theta}}_N = \boldsymbol{\theta}^{(k)}$ *and Exit*;
   - *End If*;
4. *Update.*
   - *Update*

$$\boldsymbol{\theta}^{(k+1)} = \left[\boldsymbol{\theta}^{(k)} - \boldsymbol{\eta}^{(k)} \frac{\partial^{(i,k)}}{\|\partial^{(i,k)}\|}\right]_\Theta$$

   where $\boldsymbol{\eta}^{(k)}$ *is a suitable stepsize defined in* (9);
   - *Set* $k = k + 1$ *and goto* 2.

We notice that, in this algorithm, for any candidate $\boldsymbol{\theta}^{(k)}$ the inner loop ("While'' loop in step 2) performs a randomized check of robust feasibility of the current solution. If the loop performs all the $\ell = N(k)$ checks without detecting infeasibility, then the algorithm exits at step 3 with the current solution. Otherwise, if a random sample $\mathbf{\Delta}^{(i)}$ is found such that $v(\mathbf{\Delta}^{(i)}, \boldsymbol{\theta}^{(k)}) > 0$, i.e. the performance is violated, then the current solution $\boldsymbol{\theta}^{(k)}$ is updated and the process is repeated.

Two different types of results can be proven. First, it can be shown that if the stepsizes

$$\boldsymbol{\eta}^{(k)} = \frac{v(\mathbf{\Delta}^{(k)}, \boldsymbol{\theta}^{(k)})}{\|\partial^{(i,k)}\|} + r \tag{9}$$

are chosen, and the algorithm is run with unspecified $N(k)$, then the algorithm is guaranteed to converge with probability one in a finite number of iterations to a robustly feasible solution. That is,

$$\Pr\left\{\text{There exists } \mathbf{k}_0 < \infty : \boldsymbol{\theta}^{(k)} \in \mathcal{S} \text{ for all } k \geqslant \mathbf{k}_0\right\} = 1.$$

However, this number of iterations could be in principle very high, see, for instance, [41]. This explains the introduction of the sample size function $N(k)$. In fact, if the sample size function $N(k)$ is chosen as, see [40],

$$N(k) = \overline{N}_{\mathrm{ss}}(p^*, \delta, k) \doteq \left\lceil \frac{\log \frac{\pi^2(k+1)^2}{6\delta}}{\log \frac{1}{p^*}} \right\rceil, \tag{10}$$

then, with probability greater than $1 - \delta$, if Algorithm 4 exits, the returned solution $\hat{\boldsymbol{\theta}}_N$ satisfies

$$\Pr\left\{v(\boldsymbol{\Delta}, \hat{\boldsymbol{\theta}}_N) \leqslant 0\right\} \geqslant p^*, \tag{11}$$

that is, the RA complies with Definition 3.

4.2. Stochastic ellipsoid algorithm (SEA)

An improvement upon the stochastic gradient method outlined above has been presented in [28]. This latter method produces a sequence of ellipsoids with decreasing volume, all containing the solution set $\mathcal{S}$. Provided that $\mathcal{S}$ has a nonempty interior, the centers of the ellipsoid sequence converge to a robustly feasible solution. One distinct advantage of the SEA is that it does not make use of the actual value of the feasibility radius r. Moreover, in [28] the authors prove that this algorithm also converges when the volume of $\mathcal{S}$ is zero, although convergence in this case is only asymptotic.

Ellipsoids are described by means of a center θ and a symmetric positive definite shape matrix W:

$$\mathcal{E}(\theta, W) \doteq \{x : (x - \theta)^T W^{-1}(x - \theta) \leqslant 1\}.$$

We assume that an initial ellipsoid $\mathcal{E}^{(0)} = \mathcal{E}(\theta^{(0)}, W^{(0)})$ is given such that $\mathcal{S} \subseteq \mathcal{E}^{(0)}$. For simplicity, and without loss of generality, we assume that $\Theta \equiv \mathbb{R}^{n_\theta}$. The update rule for the ellipsoid parameters is given by

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \frac{1}{n_\theta + 1} \frac{\mathbf{W}^{(k)} \partial_\theta^{(i,k)}}{\sqrt{\partial_\theta^{(i,k)T} \mathbf{W}^{(k)} \partial_\theta^{(i,k)}}}, \tag{12}$$

$$\mathbf{W}^{(k+1)} = \frac{{n_\theta}^2}{{n_\theta}^2 - 1} \left(\mathbf{W}^{(k)} - \frac{2}{n_\theta + 1} \frac{\mathbf{W}^{(k)} \partial_\theta^{(i,k)} \partial_\theta^{(i,k)T} \mathbf{W}^{(k)}}{\partial_\theta^{(i,k)T} \mathbf{W}^{(k)} \partial_\theta^{(i,k)}} \right). \tag{13}$$

Algorithm 5 (SEA for RPS).

1. *Initialization*.
   - *Let* $N(k) = \overline{N}_{\rm ss}(p^*, \delta, k)$ *given in* (10);
   - *Set* $k = 0$, $i = 0$;
   - *Choose initial ellipsoid* $\mathcal{E}^{(0)} = \mathcal{E}(\theta^{(0)}, W^{(0)}) \supseteq \mathcal{S}$;
2. *Feasibility loop*.
   - *Set* $\ell = 0$ *and* feas = true;
   - *While* $\ell < N(k)$ *and* feas = true
     - *Set* $i = i + 1$, $\ell = \ell + 1$;
     - *Draw* $\mathbf{\Delta}^{(i)}$ *according to* $f_{\mathbf{\Delta}}$;
     - *If* $v(\mathbf{\Delta}^{(i)}, \boldsymbol{\theta}^{(k)}) > 0$ *set* feas = false;
   - *End While*;
3. *Exit condition*.
   - *If* feas = true
     - *Set* $N = i$;
     - *Return* $\hat{\boldsymbol{\theta}}_N = \boldsymbol{\theta}^{(k)}$ *and Exit*;
   - *End If*;
4. *Update*.
   - *Compute* $\boldsymbol{\theta}^{(k+1)}$, $\mathbf{W}^{(k+1)}$ *according to* (12), (13);
   - *Set* $k = k + 1$ *and goto* 2.

The convergence properties of the above algorithm have been established in [40]: Algorithm 5 is guaranteed to terminate in a finite number N of total iterations. Moreover, if a lower bound V is known on the volume of the solution set $\mathrm{Vol}\,[\mathcal{S}] \geqslant V > 0$, then Algorithm 5 executes at most $\overline{k} = 2n_\theta \left\lfloor \log(\mathrm{Vol}\left[\mathcal{E}^{(0)}\right]/V) \right\rfloor$ updates, and at most

$$N \leqslant 1 + \sum_{k=0,\ldots,\overline{k}} \overline{N}_{\rm ss}(p^*, \delta, k) \leqslant 1 + (\overline{k}+1)\overline{N}_{\rm ss}(p^*, \delta, \overline{k})$$

total iterations. Upon termination, we have two cases

1. If Algorithm 5 executes $\overline{k}$ updates, then $\hat{\boldsymbol{\theta}}_N \in \mathcal{S}$, i.e. it is a robustly feasible solution.
2. With probability greater than $1-\delta$, if Algorithm 5 terminates before $\overline{k}$ updates, then $\hat{\boldsymbol{\theta}}_N$ satisfies

$$\Pr\left\{ v(\mathbf{\Delta}, \hat{\boldsymbol{\theta}}_N) \leqslant 0 \right\} \geqslant p^*$$

   i.e. it is a probabilistically feasible solution.

4.3. Other methods and related literature

Other randomized methods have been recently proposed in the literature for RPS. For instance, an improvement upon the ellipsoid algorithm is discussed in [11]. In this reference, the authors develop a sequential analytic center cutting plane method based on a probabilistic oracle that is guaranteed to converge in polynomial time to a probabilistically feasible solution. A very recent sequentially optimal RA for robust feasibility problems is proposed in [1].

A somewhat different probabilistic approach for optimization under uncertainty is instead proposed in [8]. In this reference, the authors suggest a one shot solution of a convex optimization problem with many constraints extracted at random (the scenarios), and provide a theoretical bound on the number of scenarios that need to be considered in order to guarantee the required probabilistic performance levels on the solution.

For further details on existing methods for probabilistic design the reader is referred to the books [12,52] and references therein.

5. Sample generation problem

All the previously described randomized methods critically rely on efficient techniques for random sample generation. The interested reader may refer to [19,38] for a general discussion on the topic of random number generation. In particular, in [19] several algorithms for univariate sample generation according to various distributions are shown, while in [38] Monte Carlo and quasi-Monte Carlo methods are analyzed in detail. However, no specific algorithm for vector and matrix sample generation within sets of interest in robust control is provided in the Monte Carlo literature. We also remark that standard rejection methods cannot be used, due to their inefficiency, see details in [14]. In the context of uncertain control systems described in the $M - \Delta$ form, the problem is the sample generation within $\mathbb{D}$ according to a given density function f_Δ.

For real and complex parametric uncertainties $q = [q_1 \cdots q_n]^T$, bounded in the ℓ_p norm ball of radius ρ

$$\mathcal{B}_p(\rho, \mathbb{F}^n) \doteq \left\{ q \in \mathbb{F}^n : \|q\|_p \leqslant \rho \right\}, \tag{14}$$

the sample generation problem has a simple solution. We report here an algorithm, presented in [13], that returns a real random vector $\mathbf{q}$ uniformly distributed in the norm ball $\mathcal{B}_p(\rho, \mathbb{R})$. This algorithm is based on the so-called generalized gamma density $\bar{G}_{a,c}(x)$, defined as

$$\bar{G}_{a,c}(x) = \frac{c}{\Gamma(a)} x^{ca-1} \mathrm{e}^{-x^c}, \quad x \geqslant 0,$$

where a and c are given parameters and $\Gamma(a)$ is the gamma function.

Algorithm 6 (Uniform generation in real ℓ_p norm ball). *Given n, p and ρ, the following RA returns a real random vector $\mathbf{y}$ uniformly distributed in $\mathcal{B}_p(\rho, \mathbb{R}^n)$.*

1. *Generate n independent random real scalars $\boldsymbol{\xi}_i \sim \overline{G}_{1/p,p}$;*
2. *Construct the vector $\mathbf{x} \in \mathbb{R}^n$ of components $\mathbf{x}_i = \mathbf{s}_i \boldsymbol{\xi}_i$, where $\mathbf{s}_i$ are independent random signs;*
3. *Generate $\mathbf{z} = \mathbf{w}^{1/n}$, where $\mathbf{w}$ is uniform in* [0, 1];
4. *Return $\mathbf{y} = \rho \mathbf{z} \frac{\mathbf{x}}{\|\mathbf{x}\|_p}$.*

Fig. 1 visualizes the main steps of this algorithm in the simple case of sample generation of two-dimensional real vectors in a circle of radius one ($\ell = 2$, $p = 2$, $\rho = 1$). First, we notice that for $p = 2$ the generalized gamma density $\bar{G}_{1/p,p}(x)$ is related to the Gaussian density function.

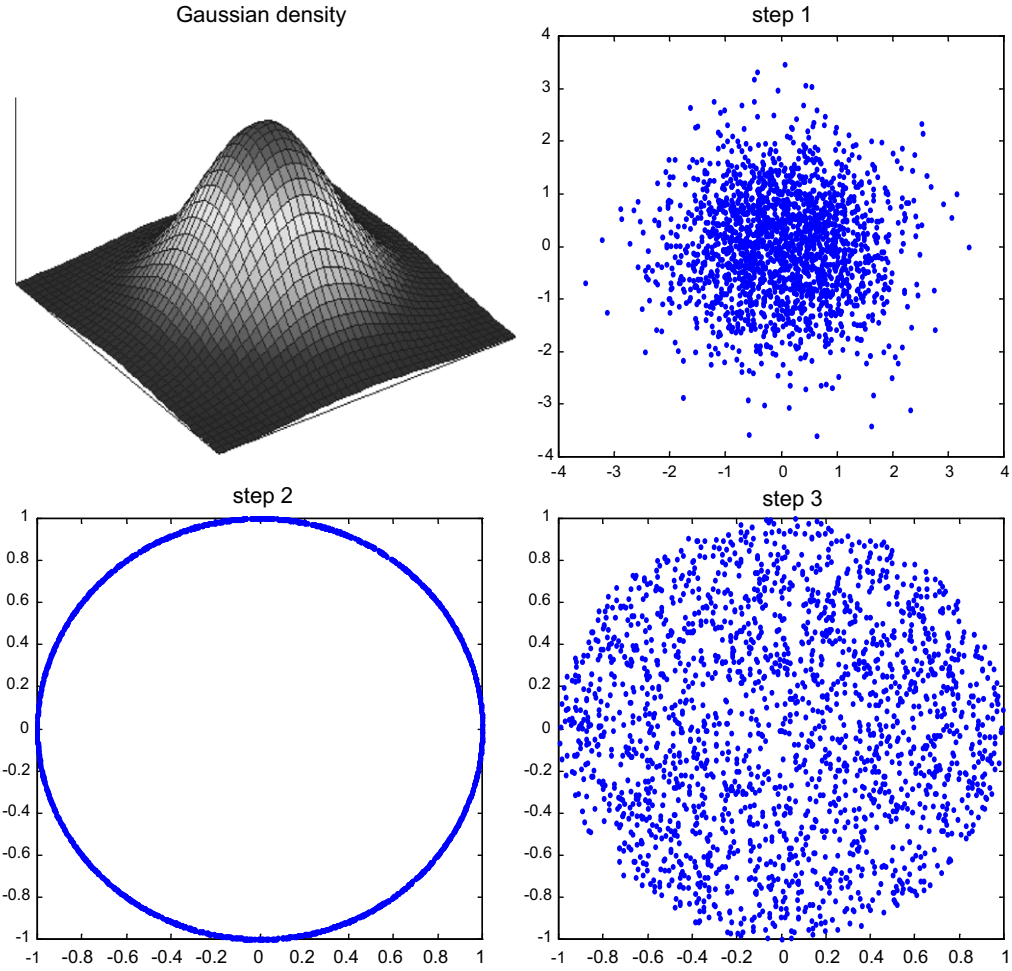

Fig. 1. Generation of real random vectors uniformly distributed in a circle.

The random samples drawn from a Gaussian distribution (step 1 in the figure) are radially symmetric with respect to the ℓ_2 norm. Roughly speaking, this means that their level curves are ℓ_2 spheres. Secondly, the samples are normalized obtaining random vectors uniformly distributed on the boundary of the circle (step 2), and then injected according to the volumetric factor z (step 3). We remark that in [52] a similar algorithm for complex vectors uniformly distributed in the norm ball $\mathcal{B}(\rho, \mathbb{C}^n)$ is presented.

The sample generation problem becomes much harder when we are interested in the uniform generation of real and complex matrix samples $\Delta \in \mathbb{F}^{n,m}$ bounded in the induced ℓ_p norm. In particular, while the cases $p = 1$ and ∞ can be immediately reduced to multiple random vector generation for which the techniques described in [13] can be used, the solution for the induced ℓ_2 norm ball requires the development of specific methods. In particular, the algorithms presented in [9,14], see also [52], are based on the singular value decomposition of the complex (real) matrix

Δ in the matrix product

$$\Delta = U\Sigma V^*,$$

where U and V are unitary (orthogonal) matrices and Σ is a diagonal matrix containing the singular values of Δ. The main idea is to compute the density functions of U, Σ and V, respectively, such that the resulting pdf $f_{\mathbf{\Delta}}(\Delta)$ is uniform. This approach is an extension of the methods described in [3,35] in the context of the theory of random matrices for the special case when Δ is a real symmetric matrix. When Δ is a causal stable dynamic operator, sampling techniques have been discussed in [10,48].

6. Conclusions

This paper is focused on nonstandard tools for analysis and control of uncertain systems, with emphasis on the interplay of probability and robustness. In this context, RAs represent a class of computationally efficient techniques that permit the solution (in probabilistic sense) of robust analysis and design problems that are otherwise hard to attack via deterministic methods. We hope that this survey will pave the way for further developments of sampling-based methods within the IBC setting.

References

[1] T. Alamo, R. Tempo, D.R. Ramirez, E.F. Camacho, A sequentially optimal randomized algorithm for robust LMI feasibility problems, in: Proceedings of the European Control Conference, 2007, to appear.
[2] T. Alpcan, T. Başar, R. Tempo, Randomized algorithms for stability and robustness of high-speed communication networks, IEEE Trans. Neural Networks 16 (2005) 1229–1241.
[3] T.W. Anderson, An Introduction to Multivariate Statistical Analysis, Wiley, New York, 1958.
[4] E.-W. Bai, R. Tempo, M. Fu, Worst-case properties of the uniform distribution and randomized algorithms for robustness analysis, Math. of Control Signals and Systems 11 (1998) 183–196.
[5] B.R. Barmish, C.M. Lagoa, The uniform distribution: a rigorous justification for its use in robustness analysis, Math. of Control Signals and Systems 10 (1997) 203–222.
[6] V.D. Blondel, J.N. Tsitsiklis, A survey of computational complexity results in systems and control, Automatica 36 (2000) 1249–1274.
[7] X. Bombois, G. Scorletti, M. Gevers, P.M.J. Van den Hof, R. Hildebrand, Least costly identification experiment for control, Automatica 42 (2006) 1651–1662.
[8] G. Calafiore, M.C. Campi, The scenario approach to robust control design, IEEE Trans. Automat. Control 51 (5) (2006) 742–753.
[9] G. Calafiore, F. Dabbene, A probabilistic framework for problems with real structured uncertainty in systems and control, Automatica 38 (2002) 1265–1276.
[10] G. Calafiore, F. Dabbene, Control design with hard/soft performance specifications: a Q-parameter randomization approach, Internat. J. Control 77 (2004) 461–471.
[11] G. Calafiore, F. Dabbene, An iterative localization method for probabilistic feasibility of uncertain LMIs, in: Proceedings of the IEEE Conference on Decision and Control, 2006.
[12] G. Calafiore, F. Dabbene (Eds.), Probabilistic and Randomized Methods for Design under Uncertainty, Springer, London, 2006.
[13] G. Calafiore, F. Dabbene, R. Tempo, Radial and uniform distributions in vector and matrix spaces for probabilistic robustness, in: D.E. Miller, L. Qiu (Eds.), Topics in Control and its Applications, Springer, New York, 1999, pp. 17–31.
[14] G. Calafiore, F. Dabbene, R. Tempo, Randomized algorithms for probabilistic robustness with real and complex structured uncertainty, IEEE Trans. Automat. Control 45 (2000) 2218–2235.
[15] G. Calafiore, B.T. Polyak, Stochastic algorithms for exact and approximate feasibility of robust LMIs, IEEE Trans. Automat. Control 46 (2001) 1755–1759.

[16] H. Chernoff, A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations, Ann. of Math. Statist. 23 (1952) 493–507.
[17] F. Dabbene, C. Lagoa, R. Tempo, Computing hard bounds for probabilistic robustness, in: Proceedings of the IEEE Conference on Decision and Control, 2006.
[18] F. Dabbene, B.T. Polyak, R. Tempo, On the complete instability of interval polynomials, Systems Control Lett. (2006), in press, doi: 10.1016/j.sysconle.2006.11.002.
[19] L.P. Devroye, Non-Uniform Random Variate Generation, Springer, New York, 1986.
[20] P. Djavdan, H.J.A.F. Tulleken, M.H. Voetter, H.B. Verbruggen, G.J. Olsder, Probabilistic robust controller design, in: Proceedings of the IEEE Conference on Decision and Control, 1989.
[21] Y. Fujisaki, F. Dabbene, R. Tempo, Probabilistic robust design of LPV control systems, Automatica 39 (2003) 1323–1337.
[22] Y. Fujisaki, Y. Oishi, Guaranteed cost regulator design: a probabilistic solution and a randomized algorithm, Automatica 43 (2007) 317–324.
[23] Y. Fujisaki, Y. Oishi, R. Tempo, A mixed probabilistic/deterministic approach to fixed order H_∞ controller design, in: Proceedings of the IEEE Conference on Decision and Control, 2006.
[24] A.J. Helmicki, C.A. Jacobson, C.N. Nett, Control-oriented system identification: a worst-case/deterministic approach in H_∞, IEEE Trans. Automat. Control AC-36 (1991) 1163–1176.
[25] W. Hoeffding, Probability inequalities for sums of bounded random variables, J. Amer. Statist. Assoc. 58 (1963) 13–30.
[26] H. Ishii, T. Başar, R. Tempo, Randomized algorithms for quadratic stability of quantized sampled-data systems, Automatica 40 (2004) 839–846.
[27] H. Ishii, T. Başar, R. Tempo, Randomized algorithms for synthesis of switching rules for multimodal systems, IEEE Trans. Automat. Control 50 (2005) 754–767.
[28] S. Kanev, B. De Schutter, M. Verhaegen, An ellipsoid algorithm for probabilistic robust controller design, Systems Control Lett. 49 (2003) 365–375.
[29] P. Khargonekar, A. Tikku, Randomized algorithms for robust control analysis and synthesis have polynomial complexity, in: Proceedings of the IEEE Conference on Decision and Control, 1996.
[30] V. Koltchinskii, C.T. Abdallah, M. Ariola, P. Dorato, D. Panchenko, Improved sample complexity estimates for statistical learning control of uncertain systems, IEEE Trans. Automat. Control 46 (2000) 2383–2388.
[31] H.J. Kushner, G.G. Yin, Stochastic Approximation Algorithms and Applications, Springer, New York, 1997.
[32] C.M. Lagoa, B.R. Barmish, Distributionally robust Monte Carlo simulation: a tutorial survey, in: Proceedings of the IFAC World Congress, 2002.
[33] D. Liberzon, R. Tempo, Common Lyapunov functions and gradient algorithms, IEEE Trans. Automat. Control 49 (2004) 990–994.
[34] L. Lorefice, B. Pralio, R. Tempo, Robust and randomized control design of mini-UAVs: the MH1000 platform, in: Proceedings of the American Control Conference, 2007.
[35] M.L. Mehta, Random Matrices, Academic Press, Boston, 1991.
[36] M. Milanese, R. Tempo, Optimal algorithms theory for robust estimation and prediction, IEEE Trans. Automat. Control AC-30 (1985) 730–738.
[37] A. Nemirovski, Several NP-hard problems arising in robust stability analysis, Math. of Control Signals and Systems 6 (1993) 99–195.
[38] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, SIAM, Philadelphia, PA, 1992.
[40] Y. Oishi, Polynomial-time algorithms for probabilistic solutions of parameter-dependent linear matrix inequalities, Automatica 43 (2007) 538–545.
[41] Y. Oishi, H. Kimura, Computational complexity of randomized algorithms for solving parameter-dependent linear matrix inequalities, Automatica 39 (2003) 2149–2156.
[42] L. Palopoli, C. Pinello, A. Bicchi, A. Sangiovanni-Vincentelli, Maximizing the stability radius of a set of systems under real-time scheduling constraints, IEEE Trans. Automat. Control 50 (2005) 1790–1795.
[43] S. Poljak, J. Rohn, Checking robust nonsingularity is NP-hard, Math. of Control Signals and Systems 6 (1993) 1–9.
[44] D. Pollard, Convergence of Stochastic Processes, Springer, New York, 1984.
[45] B.T. Polyak, R. Tempo, Probabilistic robust design with linear quadratic regulators, Systems Control Lett. 43 (2001) 343–353.
[46] R.F. Stengel, Some effects of parameter variations on the lateral-directional stability of aircraft, AIAA J. Guidance and Control 3 (1980) 124–131.
[47] R.F. Stengel, Stochastic Optimal Control: Theory and Application, Wiley, New York, 1986.
[48] M. Sznaier, C. Lagoa, M.C. Mazzaro, An algorithm for sampling subsets of H_∞ with applications to risk-adjusted performance analysis and model (in)validation, IEEE Trans. Automat. Control 50 (2005) 410–416.

[49] R. Tempo, Robust estimation and filtering in the presence of bounded noise, IEEE Trans. Automat. Control AC-33 (1988) 864–867.
[50] R. Tempo, E.-W. Bai, F. Dabbene, Probabilistic robustness analysis: explicit bounds for the minimum number of samples, in: Proceedings of the IEEE Conference on Decision and Control, 1996.
[51] R. Tempo, E.-W. Bai, F. Dabbene, Probabilistic robustness analysis: explicit bounds for the minimum number of samples, Systems Control Lett. 30 (1997) 237–242.
[52] R. Tempo, G. Calafiore, F. Dabbene, Randomized Algorithms for Analysis and Control of Uncertain Systems, Communications and Control Engineering Series, Springer, London, 2005.
[53] R. Tempo, H. Ishii, Monte Carlo and Las Vegas randomized algorithms for systems and control: an introduction, European J. of Control (2007), to appear.
[54] J.F. Traub, G.W. Wasilkowski, H. Woźniakowski, Information-Based Complexity, Academic Press, New York, 1988.
[55] J.F. Traub, A.G. Werschulz, Complexity and Information, Cambridge University Press, Cambridge, 1998.
[56] J.F. Traub, H. Woźniakowski, A General Theory of Optimal Algorithms, Academic Press, New York, 1980.
[57] J.F. Traub, H. Woźniakowski, Breaking intractability, Sci. Amer. 270 (1994) 102–107.
[58] V.A. Ugrinovskii, Randomized algorithms for robust stability and guaranteed cost control of stochastic jump parameter systems with uncertain switching policies, J. Optim. Theory Appl. 124 (2005) 227–245.
[59] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[60] M. Vidyasagar, Statistical learning theory and randomized algorithms for control, IEEE Control Systems Mag. 18 (1998) 69–85.
[61] M. Vidyasagar, Randomized algorithms for robust controller synthesis using statistical learning theory, Automatica 37 (2001) 1515–1528.
[62] M. Vidyasagar, Learning and Generalization: With Applications to Neural Networks, second ed., Springer, New York, 2002.
[63] M. Vidyasagar, V. Blondel, Probabilistic solutions to some NP-hard matrix problems, Automatica 37 (2001) 1397–1405.
[64] G. Zames, On the metric complexity of causal linear systems: ε-entropy and ε-dimension for continuous-time systems, IEEE Trans. Automat. Control AC-24 (1979) 222–230.
[65] G. Zames, Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms and approximate inverses, IEEE Trans. Automat. Control 26 (1981) 301–320.