PERGAMON

Automatica 38 (2002) 1265–1276

automatica

www.elsevier.com/locate/automatica

A probabilistic framework for problems with real structured uncertainty in systems and control☆

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

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

Received 23 March 2000; received in revised form 26 July 2001; accepted 14 January 2002

Abstract

The objective of this paper is twofold. First, the problem of generation of real random matrix samples with uniform distribution in structured (spectral) norm bounded sets is studied. This includes an analysis of the distribution of the singular values of uniformly distributed real matrices, and an efficient (i.e. polynomial-time) algorithm for their generation. Second, it is shown how the developed techniques may be used to solve in a probabilistic setting several hard problems involving systems subject to real structured uncertainty.

Keywords: Randomized algorithms; Robustness; Structured real uncertainty

1. Introduction

In this paper we consider a broad class of problems involving static or dynamic systems subject to structured and norm bounded real uncertainty. Problems of this kind arise in many contexts, such as robust analysis and control of LTI systems (Qiu et al., 1995; Zhou, Doyle & Glover, 1996), estimation and filtering (El Ghaoui & Calafiore, 2001; Xie, Soh, & de Souza, 1994), robust optimization (Ben-Tal, El Ghaoui, & Nemirovskii, 2000; El Ghaoui, Oustry, & Lebret, 1998), etc. Several results are now available showing that such problems can be computationally hard to solve exactly (Blondel & Tsitsiklis, 2000; Nemirovskii, 1993; Polijak & Rohn, 1993), therefore a variety of methods have been proposed to solve relaxed versions of the original problems, at the expense of possible conservatism (Zhou et al., 1996; Zhu, Huang, & Doyle, 1996).

Recently, a new parallel line of research has emerged, which proposes to substitute the robust or worst-case viewpoint with a more tractable probabilistic one (Ray & Stengel, 1993; Tempo & Dabbene, 1999; Vidyasagar & Blondel, 2001; Vidyasagar, 2001). It turns out that problems that are computationally hard in a deterministic setting may be efficiently solved using randomized algorithms, if a certain probability of performance degradation is accepted (Khargonekar & Tikku, 1996; Tempo, Bai, & Dabbene, 1997; Vidyasagar, 1997). For these reasons, a now generally accepted statement is that "randomized algorithms have polynomial-time complexity". A crucial remark, however, is that the previous statement is true provided that the computational cost required to generate random samples of the uncertainty according to the required probability distribution is indeed polynomial-time. Unfortunately, no such algorithm was available for the most common uncertainty structure, where the uncertainty matrix Δ is block-diagonal with blocks Δ_i bounded in the spectral norm. The difficulty of this sample generation problem may be easily overlooked, therefore a brief discussion on the failing of first-attempt techniques is reported in Section 2. The uncertainty sample generation problem for the case of complex matrix blocks was studied in depth in Calafiore, Dabbene, and Tempo (2000a), while the case of vector uncertainty has been treated in Calafiore, Dabbene, and Tempo (1999a). However, the sample generation problem in the case of real matrix blocks raised further technical difficulties that could not be overcome within the technical framework introduced in Calafiore et al. (2000a), and was therefore left as an open problem.

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor R. Srikant under the direction of Editor Tamer Basar. This work was supported in part by CNR Agenzia 2000 funds.

* Corresponding author.
E-mail addresses: calafiore@polito.it (G. Calafiore), dabbene@polito.it (F. Dabbene).

0005-1098/02/$ - see front matter © 2002 Elsevier Science Ltd. All rights reserved.
PII: S0005-1098(02)00015-8

The main objective of this paper is to develop the mathematical techniques required for the solution of the real matrix blocks sample generation problem, and to show how these techniques may be applied to solve in a probabilistic setting several hard problems arising in systems and control.

In particular, in this paper we will assume that the uncertainty affecting the system has the structure described by the set

$$\mathbf{\Delta} \doteq \{\Delta : \Delta = \mathrm{diag}(q_1 I_{r_1}, q_s I_{r_s}, \Delta_1, \ldots, \Delta_b)\}, \tag{1}$$

where $q_i \in \mathbb{R}$, $i=1,\ldots,s$ are repeated scalar parameters with multiplicity $r_1,\ldots,r_s$, and $\Delta_i \in \mathbb{R}^{n_i,m_i}$, $i=1,\ldots,b$ are full blocks. The structured matrix Δ is further restricted to the set

$$\mathbf{\Delta}_\rho \doteq \{\Delta \in \mathbf{\Delta} : \|\Delta\|_\sigma \leqslant \rho\}, \tag{2}$$

where $\|\Delta\|_\sigma \doteq \bar{\sigma}(\Delta)$, and $\bar{\sigma}(\cdot)$ denotes the maximum singular value of a matrix. The set $\mathbf{\Delta}_\rho$ is the subset of perturbations in $\mathbf{\Delta}$ with size at most ρ. This uncertainty description is general and it is now widely used in the context of uncertain systems, see for instance (Zhou et al., 1996) and references therein. We will make the standing assumption that the blocks Δ_i are independent random matrices having uniform distribution on the spectral norm bounded support

$$\mathscr{B}_\rho \doteq \{\Delta \in \mathbb{R}^{n,m} : \bar{\sigma}(\Delta) \leqslant \rho\}. \tag{3}$$

The scalars q_i are also considered independent and uniformly distributed; their generation is trivial and will not be further discussed.

A justification of the choice of the uniform density may be found for instance in Bai, Tempo, and Fu (1998) and Barmish and Lagoa (1997). However, the results in this paper hold not only for the uniform distribution, but for the more general class of unitarily invariant distributions over $\mathscr{B}_\rho$, see (Calafiore et al., 2000a). The independence assumption reduces the problem of generating samples of uncertainty in the set $\mathbf{\Delta}_\rho$ to the problem of generating a single block in the set $\mathscr{B}_\rho$. Therefore, in the sequel of the paper we focus, without loss of generality, to the problem of generating uniform samples of $\Delta \in \mathscr{B}_\rho$. This problem has also its own theoretical interest in the theory of multivariate statistical analysis, and is thoroughly studied in the paper.

The first part of the paper is technical, and discusses a method for generating uniform samples of $\Delta \in \mathscr{B}_\rho$ based on the generation of the singular value decomposition (SVD) factors of Δ. In Section 2, the density function of the singular values of uniformly distributed matrices is studied, and a polynomial-time algorithm for their generation is presented in Section 2.2.

In Section 3, we show how several problems involving systems subject to real structured uncertainty may be efficiently solved in a probabilistic setting, and discuss some applications to selected problems. In particular, we consider the solution of uncertain least-squares problems in Section 3.1, the computation of the structured real stability radius in Section 3.2, and the assessment of approximate feasibility in robust semidefinite programming (SDP) in Section 3.3. An application to robust control design is also studied in Calafiore, Dabbene, and Tempo (2000b).

1.1. Definitions and notation

The space of real skew-symmetric matrices of order n will be denoted as $\mathscr{S}^n \doteq \{X \in \mathbb{R}^{n,n} : X + X^{\mathrm{T}} = 0\}$. The determinant of a real square matrix X is denoted by either $|X|$ or $\det X$, and $\mathrm{Adj}\, X$ denotes the classical adjoint of X. When needed by the context, we specify the dimension of the identity and the zero matrix with I_n and $0_{n,m}$.

A real random matrix $\Delta \in \mathbb{R}^{n,m}$ is a matrix of random variables $[\Delta]_{i,k}$, $i=1,\ldots,n$; $k=1,\ldots,m$. The probability density function (pdf) $f_\Delta(\Delta)$ is defined as the joint probability density of the elements of Δ. The notation $\Delta \sim f_\Delta$ means that Δ is a random matrix with probability density f_Δ. For a measurable set $\mathscr{F} \subset \mathbb{R}^{n,m}$, $\Delta \sim \mathscr{U}[\mathscr{F}]$ means that Δ is a random matrix with uniform density over the set $\mathscr{F}$.

Given a vector $[x_1, x_2, \ldots, x_n]^{\mathrm{T}}$, its Vandermonde matrix $\mathscr{V}_n$ is defined as $\mathscr{V}_n = \mathscr{V}_n(x_1,\ldots,x_n) \doteq [\mathscr{X}(x_1)\ \mathscr{X}(x_2)\ \cdots\ \mathscr{X}(x_n)]$, $\mathscr{X}(\xi) \doteq [1\ \xi\ \xi^2\ \cdots\ \xi^{n-1}]^{\mathrm{T}}$. Similarly, $\mathscr{V}_i = \mathscr{V}_i(x_1, x_2, \ldots, x_i)$ is defined as the truncated Vandermonde matrix composed of the first i columns of $\mathscr{V}_n$.

Definition 1 (Normalized SVD). Given $\Delta \in \mathbb{R}^{n,m}$, $m \geqslant n$, we define the following normalized form of the SVD of Δ: $\Delta = U\Sigma V^{\mathrm{T}}$, where $U \in \mathbb{R}^{n,n}$ and $V \in \mathbb{R}^{m,n}$ have orthonormal columns, and $\Sigma = \mathrm{diag}(\sigma_1,\ldots,\sigma_n)$, with $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0$. The columns of V are normalized so that the first nonvanishing component of each column is positive.

2. Sample generation in a spectral norm bounded set

The aim of this first section is to provide the technical details for the solution of the problem of generating samples of a matrix $\Delta \in \mathbb{R}^{n,m}$ uniformly distributed in the spectral norm ball of radius one, i.e. $\Delta \sim \mathscr{U}[\mathscr{B}_1]$. Uniform distribution in a set of generic radius ρ may be easily obtained multiplying by ρ the samples generated in the unit ball.

Due to dimensionality problems, classical generation methods based on rejection of samples from an outer bounding set (Devroye, 1986) are highly inefficient for the problem at hand, as already remarked in Calafiore et al. (2000a) for the case of complex matrices. Indeed, the *rejection rate* γ—defined as the expected number of samples that should be generated in the outer bounding set in order to find one sample in the desired set—increases exponentially with the dimension n. Let for instance $\Delta \in \mathbb{R}^{n,n}$, and consider the sets $\mathscr{A}_{\mathrm{F}} \doteq \{\Delta : \|\Delta\|_{\mathrm{F}} \leqslant \sqrt{n}\}$, and $\mathscr{A}_{\mathrm{C}} \doteq \{\Delta : \max_{i=1,\ldots,n} \|\delta_i\| \leqslant 1\}$, where the subscript F stands for Frobenius norm, and δ_i, $i=1,\ldots,n$, represent the columns of Δ. Then, we have the inclusions $\mathscr{B}_1 \subseteq \mathscr{A}_{\mathrm{F}}$, and $\mathscr{B}_1 \subseteq \mathscr{A}_{\mathrm{C}}$. In Table 1, we report the rejection rates for samples uniformly generated in the above defined sets, computed as

Table 1
Rejection rates γ_F from the set $\mathscr{A}_F$, and γ_C from the set $\mathscr{A}_C$

n	2	3	4	5	6	8	10	12
γ_F	3	26.7	640	3.9e4	6.1e6	2.3e12	3.1e19	1.54e28
γ_C	1.5	4.24	24.6	305	8.3e3	6.9e7	1.5e13	9.7e19

the ratio between the volumes of the outer bounding and the desired sets. The required formulas for the volumes have been derived in Calafiore et al. (1999a, 2000a).

Table 1 shows the inefficiency of methods based on rejection, due to the "curse of dimensionality", and motivates the need for more sophisticated techniques for direct generation of samples in $\mathscr{B}_1$. To this end, the key idea is to construct the samples of Δ from the samples of the SVD factors U, Σ, V defined in Definition 1.

The relation among the uniform pdf of Δ and the pdfs of U, Σ, V was studied in Calafiore et al. (2000a). That key result is reported below for easier reference of the reader.

Theorem 1 (Calafiore et al., 2000a). *Let $\Delta \in \mathbb{R}^{n,m}$, $m \geqslant n$, be factored as in Definition* 1, *with* $1 > \sigma_1 > \cdots > \sigma_n > 0$. *The following statements are equivalent*:

(i) $\Delta \sim \mathscr{U}[\mathscr{B}_1]$;
(ii) *The joint pdf of U, Σ and V is* $f_{U,\Sigma,V}(U,\Sigma,V) = f_U(U) f_\Sigma(\Sigma) f_V(V)$, *where* $f_U(U) = \mathscr{U}[\{U : U^{\mathrm{T}}U = I\}]$, $f_V(V) = \mathscr{U}[\{V : V^{\mathrm{T}}V = I, [V]_{1,j} > 0,\ j = 1,\ldots,n\}]$, *and*

$$f_\Sigma(\Sigma) = K_{\mathbb{R}} \prod_{j=1}^{n} \sigma_j^{m-n} \prod_{1 \leqslant j < k \leqslant n} (\sigma_j^2 - \sigma_k^2) \tag{4}$$

and $K_{\mathbb{R}}$ is a normalization constant given by

$$K_{\mathbb{R}} = n!\pi^{n/2} \prod_{j=0}^{n-1} \frac{\Gamma(1 + \frac{m+j}{2})}{\Gamma(\frac{3}{2} + \frac{j}{2})\Gamma(\frac{m-n+j+1}{2})\Gamma(1 + \frac{j}{2})}. \tag{5}$$

The value of $K_{\mathbb{R}}$ is determined using the techniques described in Selberg (1944).

In the literature, the uniform distribution over the orthogonal group is known as the Haar invariant distribution (Anderson, 1984). Further discussion and the proof of the previous result may be found in Calafiore et al. (2000a).

This result provides explicit expressions of the (independent) pdfs of U, Σ, V. We can therefore generate uniform samples of Δ by generating independently samples of U, V and Σ according to their pdfs, and then computing the product $U\Sigma V^{\mathrm{T}}$. The problem of generating U and V has been thoroughly discussed in Calafiore et al. (2000a) and in Stewart (1980). The problem of efficiently generating the singular values according to their pdf (4) is an open problem and is addressed in the next section.

2.1. On the marginal densities of the singular values

In this section, we focus on the generation of random samples of the singular values of a matrix $\Delta \in \mathscr{B}_1 \subset \mathbb{R}^{n,m}$, according to pdf (4). To this aim, we make use of a standard method for random generation with multivariate distributions, the so-called conditional method (Devroye, 1986). The basic idea of this method is to rewrite a multivariate density function as a product of conditional densities. One can then generate the first random variable according to its (univariate) marginal density, then generate the next variable conditional on the first one, and so forth. In other words, the conditional method reduces an n-dimensional generation problem to n one-dimensional problems. In our case, we have

$$f_\Sigma(\Sigma) = f_{\sigma_1}(\sigma_1) f_{\sigma_2}(\sigma_2|\sigma_1) \cdots f_{\sigma_n}(\sigma_n|\sigma_1 \cdots \sigma_{n-1}),$$

where the conditional densities $f_{\sigma_i}(\sigma_i|\sigma_1,\ldots,\sigma_{i-1})$ are given by the ratio of marginal densities

$$f_{\sigma_i}(\sigma_i|\sigma_1,\ldots,\sigma_{i-1}) = \frac{f_\sigma^{(i)}(\sigma_1,\ldots,\sigma_i)}{f_\sigma^{(i-1)}(\sigma_1,\ldots,\sigma_{i-1})}. \tag{6}$$

In turn, the marginal densities $f_\sigma^{(i)}(\sigma_1,\ldots,\sigma_i)$, $i = 1,\ldots,n$ are defined as the multiple integral

$$f_\sigma^{(i)}(\sigma_1,\ldots,\sigma_i) = \int \cdots \int f_\Sigma(\Sigma)\, \mathrm{d}\sigma_{i+1} \cdots \mathrm{d}\sigma_n. \tag{7}$$

A singular values matrix Σ with density $f_\Sigma(\Sigma)$ can therefore be obtained generating sequentially the σ_i's, $i = 1,\ldots,n$, where each σ_i is distributed according to the univariate density $f_{\sigma_i}(\sigma_i|\sigma_1,\ldots,\sigma_{i-1})$. However, this method requires the computation of the marginal densities (7), which is often a very difficult task (Devroye, 1997). To address this problem, we first state the following lemma.

Lemma 1. *The marginal densities* (7) *may be written in the form*

$$f_\sigma^{(i)}(\sigma_1,\ldots,\sigma_i) = \frac{K_{\mathbb{R}}}{2^{n-i}}\, \Upsilon_i(\sigma_1^2,\ldots,\sigma_i^2) \prod_{k=1}^{i} \sigma_k^{m-n}, \tag{8}$$

where

$$\Upsilon_i = \Upsilon_i(x_1,\ldots,x_i) \doteq \int_{\mathscr{D}_i} |\mathscr{V}_n|\, \mathrm{d}\mu(x_n) \cdots \mathrm{d}\mu(x_{i+1}), \tag{9}$$

being $\mathrm{d}\mu(x_k) \doteq x_k^\nu\, \mathrm{d}x_k$, $\nu = \frac{1}{2}(m - n - 1)$, *and* $\mathscr{D}_i \doteq \{0 < x_n < \cdots < x_i\}$.

The proof of this lemma is reported in Appendix A. We now concentrate on the closed form solution of the multiple

integral (9). The following key theorem provides a direct way to compute the conditional densities that are needed for the application of the conditional method.

Theorem 2. *For $i=1,\dots,n$, the multiple integral in* (9) *can be computed as*

$$\Upsilon_i = x_i^{\alpha_i} \det{}^{1/2} \left[\begin{array}{c|c} M(x_i) & \begin{matrix}\mathscr{V}_{i-1}\\ 0\end{matrix} \\ \hline -\mathscr{V}_{i-1}^{\mathrm{T}}\ 0 & 0 \end{array}\right], \tag{10}$$

where the blocks corresponding to the term $\mathscr{V}_{i-1} = \mathscr{V}_{i-1}(x_1,\dots,x_{i-1})$ are not present if $i=1$, and $\alpha_i \doteq (v+1)(n-i)$,

$$M(x_i) \doteq \begin{cases} \begin{bmatrix} \bar{S}(x_i) & \mathscr{X}(x_i) \\ -\mathscr{X}^{\mathrm{T}}(x_i) & 0 \end{bmatrix} & \text{if } (n-i) \text{ even}, \\ \begin{bmatrix} \bar{S}(x_i) & \mathscr{X}(x_i) & \bar{F}(x_i) \\ -\mathscr{X}^{\mathrm{T}}(x_i) & 0 & 0 \\ -\bar{F}^{\mathrm{T}}(x_i) & 0 & 0 \end{bmatrix} & \text{if } (n-i) \text{ odd}, \end{cases} \tag{11}$$

and, for $r, j = 1,\dots,n$,

$$\bar{S}_{rj}(x_i) \doteq \frac{r-j}{(r+v)(j+v)(r+j+2v)} x_i^{r+j-2}, \qquad \bar{F}_j(x_i) \doteq \frac{x_i^{j-1}}{j+v}, \qquad \mathscr{X}_j(x_i) \doteq x_i^{j-1}. \tag{12}$$

The proof of this result is lengthy and requires the introduction of further preliminary concepts. The complete proof is reported in Appendix C.

Remark 1. Notice that the matrix appearing in (10) is a skew-symmetric polynomial matrix of even order, therefore its determinant is always a perfect square in the entries of the matrix, see e.g. Vein and Dale (1999). Considering the factor $x_i^{\alpha_i}$ in (10), and recalling that $v=(m-n-1)/2$, it is straightforward to verify that $\Upsilon_i(\sigma_1^2,\dots,\sigma_i^2)$ is a multivariate polynomial in $\sigma_1,\dots,\sigma_i$. At the ith step of the conditional method, the variables $\sigma_1,\dots,\sigma_{i-1}$ are assigned to numerical values, therefore the conditional density $f_{\sigma_i}(\sigma_i|\sigma_1,\dots,\sigma_{i-1})$ defined in (6) is a univariate polynomial in σ_i. Sample generation according to a given univariate polynomial density is a standard problem, and can be efficiently performed using one of the available methods, see e.g. Devroye (1986).

2.2. An efficient algorithm for the generation of the singular values

In this section, we propose an efficient algorithm for computing recursively the marginal densities given in (8), using the results of Theorem 2. We first state a lemma that provides a direct way to compute the square root of the determinant of real skew-symmetric matrices obtained adding two rows and two columns to a given real skew-symmetric matrix of even order. Formally, let q be an even integer and, for $k=1,\dots,q/2$, let $\bar{H}_{2k}(x) \in \mathbb{R}^{q+2(k-1),2}$ be a polynomial matrix in the scalar variable x, with

$$\bar{H}_{2k}(x) \doteq \begin{bmatrix} H_{2k}(x) \\ 0_{2(k-1),2} \end{bmatrix}, \quad H_{2k}(x) \doteq [h_{2k-1}(x) \quad h_{2k}(x)], \tag{13}$$

where $h_{2k-1}(x), h_{2k}(x) \in \mathbb{R}^{q,1}$ are given polynomial vectors in x. Let also $Q_0(x) \in \mathscr{S}^q$ be a given skew-symmetric polynomial matrix, and define, for $k=1,\dots,q/2$, the bordered matrix

$$Q_{2k}(x) \doteq \left[\begin{array}{c|c} Q_{2(k-1)}(x) & \bar{H}_{2k}(x) \\ \hline -\bar{H}_{2k}^{\mathrm{T}}(x) & 0_{2,2} \end{array}\right] \in \mathscr{S}^{q+2k} \tag{14}$$

and, for $k=0,\dots,q/2$,

$$p_{2k}(x) \doteq \sqrt{|Q_{2k}(x)|}. \tag{15}$$

Notice that, from the properties of determinants of skew-symmetric matrices it follows that $p_{2k}(x)$ is a polynomial in x, see for instance Vein and Dale (1999). We are now ready to state the following lemma, which plays a key role for the development of the sample generation algorithm.

Lemma 2. *The polynomial $p_{2k}(x)$ defined in* (15) *may be computed* (*up to a sign*) *as*

$$p_{2k}(x) = \pm \frac{d_{2k}(x)}{p_0^{2k-1}(x)} \quad \text{for } k=1,\dots,q/2, \tag{16}$$

where $p_0(x) = \sqrt{|Q_0(x)|}$, and $d_{2k}(x)$ is a polynomial determined according to the following recursion: *for $k=1,\dots,q/2$,*

$$d_{2k}(x) = h_{2k-1}^{\mathrm{T}}(x) Z_{2(k-1)}(x) h_{2k}(x), \tag{17}$$

$$Z_{2k}(x) = d_{2k}(x) Z_{2(k-1)}(x) + Z_{2(k-1)}(x)(H_{2k}(x) J H_{2k}(x)^{\mathrm{T}}) Z_{2(k-1)}(x), \tag{18}$$

where $Z_0(x) = \mathrm{Adj}\, Q_0(x)$,

$$J \doteq \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

and H_{2k}, h_{2k-1}, h_{2k} are defined in (13).

The proof of this lemma is reported in Appendix B.

The result of Lemma 2 is now used to develop an efficient recursive algorithm for the generation of the singular values. In order to apply Lemma 2 to the result of Theorem 2, four different cases need to be considered, corresponding to the combinations of n and i being even or odd. This is necessary

since Lemma 2 requires the use of matrices of even order. We explain the use of the lemma for the case n even, the case n odd follows from a similar reasoning. Consider the case n even: we construct two parallel recursions, one that holds for i even, and one for i odd. If i is even, then the determinant in (10) may be immediately rewritten in form (14), with $i = 2k$,

$$Q_0(x_i) = \bar{S}(x_i), \qquad H_{2k} = [X(x_{2k-1}) \quad X(x_{2k})].$$

Thus, recursion (17) provides the Υ_i's for i even. If i is odd, then the determinant in (10) may still be rewritten in the form of Lemma 2, shifting the indices $k \leftarrow k + 1/2$ in (14)

$$Q_1(x_i) = \begin{bmatrix} \bar{S}(x_i) & \bar{F}(x_i) & \mathscr{X}(x_i) \\ -\bar{F}^{\mathrm{T}}(x_i) & 0 & 0 \\ -\mathscr{X}^{\mathrm{T}}(x_i) & 0 & 0 \end{bmatrix},$$

$$H_{2k+1} = \begin{bmatrix} X(x_{2k}) & X(x_{2k+1}) \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Recursion (17) now provides the Υ_i's for i odd.

We report below the algorithm for the generation of the singular values. In the algorithm, we denote with x and σ the current variables, while the subscripted terms x_i, σ_i denote variables evaluated to their numerical values.

Algorithm. *Define the following quantities*:

$$Q_0^{(\mathrm{e})}(x) = \bar{S}(x), \quad Q_1^{(\mathrm{e})}(x) = \begin{bmatrix} \bar{S}(x) & \bar{F}(x) & \mathscr{X}(x) \\ -\bar{F}^{\mathrm{T}}(x) & 0 & 0 \\ -\mathscr{X}^{\mathrm{T}}(x) & 0 & 0 \end{bmatrix},$$

$$Q_0^{(\mathrm{o})}(x) = \begin{bmatrix} \bar{S}(x) & \bar{F}(x) \\ -\bar{F}^{\mathrm{T}}(x) & 0 \end{bmatrix}, \quad Q_1^{(\mathrm{o})}(x) = \begin{bmatrix} \bar{S}(x) & \mathscr{X}(x) \\ -\mathscr{X}^{\mathrm{T}}(x) & 0 \end{bmatrix},$$

where $\bar{S}$, $\bar{F}$, *and* $\mathscr{X}$ *are defined in* (12). *A random matrix* $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ *distributed according to* (4), *can be generated via the following algorithm.*

Initialization (see Remark 2).

- If n is even, then

$$Z_0(x) = \mathrm{Adj}\, Q_0^{(\mathrm{e})}(x), \quad Z_1(x) = \mathrm{Adj}\, Q_1^{(\mathrm{e})}(x)$$
$$p_0(x) = \sqrt{|Q_0^{(\mathrm{e})}(x)|}, \quad p_1(x) = \sqrt{|Q_1^{(\mathrm{e})}(x)|}.$$

- If n is odd, then

$$Z_0(x) = \mathrm{Adj}\, Q_0^{(\mathrm{o})}(x), \quad Z_1(x) = \mathrm{Adj}\, Q_1^{(\mathrm{o})}(x)$$
$$p_0(x) = \sqrt{|Q_0^{(\mathrm{o})}(x)|}, \quad p_1(x) = \sqrt{|Q_1^{(\mathrm{o})}(x)|}.$$

- $\Upsilon_1(x) = x^{(v+1)(n-1)} p_1(x)$.
- $f_\sigma^{(1)}(\sigma) = K_{\mathbb{R}}/(2^{n-1}) \Upsilon_1(\sigma^2) \sigma^{m-n}$.
- Generate σ_1 according to $f_{\sigma_1}(\sigma) \equiv f_\sigma^{(1)}(\sigma)$. Set $x_1 = \sigma_1^2$.
- $k = 1$, if $n = 1$ then go to [End].

Generation (even).

- If n even, $h(x) = \mathscr{X}(x)$, else $h(x) = \left[\begin{smallmatrix} \mathscr{X}(x) \\ 0 \end{smallmatrix}\right]$.
- $d_{2k}(x) = h^{\mathrm{T}}(x_{2k-1}) Z_{2(k-1)}(x) h(x)$.
- $p_{2k}(x) = d_{2k}(x)/p_0^{2k-1}(x)$.
- $\Upsilon_{2k}(x) = \Upsilon_{2k}(\sigma_1^2, \ldots, \sigma_{2k-1}^2, x) = x^{(v+1)(n-2k)} p_{2k}(x)$.
- $f_\sigma^{(2k)}(\sigma) = f_\sigma^{(2k)}(\sigma | \sigma_1, \ldots, \sigma_{2k-1})$

$$= \pm (K_{\mathbb{R}}/(2^{n-2k})) \Upsilon_{2k}(\sigma^2) \sigma^{m-n} \prod_{j=1}^{2k-1} \sigma_j^{m-n}.$$

- Generate σ_{2k} according to $f_{\sigma_{2k}}(\sigma) = f_\sigma^{(2k)}(\sigma)/f_\sigma^{(2k-1)}(\sigma_1, \ldots, \sigma_{2k-1})$. Set $x_{2k} = \sigma_{2k}^2$.

Update (even).

- $d_{2k}(x) = h^{\mathrm{T}}(x_{2k-1}) Z_{2(k-1)}(x) h(x_{2k})$.
- $Z_{2k}(x) = d_{2k}(x) Z_{2(k-1)}(x) + Z_{2(k-1)}(x) \cdot \times [h(x_{2k-1}) h(x_{2k})] J [h(x_{2k-1}) h(x_{2k})]^{\mathrm{T}} Z_{2(k-1)}(x)$.
- If $2k + 1 > n$, then goto [End].

Generation (odd).

- If n even, $h(x) = \begin{bmatrix} \mathscr{X}(x) \\ 0 \end{bmatrix}$, else $h(x) = \begin{bmatrix} \mathscr{X}(x) \\ 0 \\ 0 \end{bmatrix}$.
- $d_{2k+1}(x) = h^{\mathrm{T}}(x_{2k-1}) Z_{2k-1}(x) h(x_{2k})$.
- $p_{2k+1}(x) = d_{2k+1}(x)/p_1^{2k-1}(x)$.

$$\Upsilon_{2k+1}(x) = \Upsilon_{2k+1}(\sigma_1^2, \ldots, \sigma_{2k}^2, x)$$
$$= x^{(v+1)(n-2k-1)} p_{2k+1}(x).$$

$$f_\sigma^{(2k+1)}(\sigma) = f_\sigma^{(2k+1)}(\sigma | \sigma_1, \ldots, \sigma_{2k})$$
$$= \pm (K_{\mathbb{R}}/(2^{n-2k-1})) \Upsilon_{2k+1}(\sigma^2) \sigma^{m-n} \times \prod_{j=1}^{2k} \sigma_j^{m-n}.$$

- Generate σ_{2k+1} according to $f_{\sigma_{2k+1}}(\sigma) = f_\sigma^{(2k+1)}(\sigma)/f_\sigma^{(2k)}(\sigma_1, \ldots, \sigma_{2k})$. Set $x_{2k+1} = \sigma_{2k+1}^2$.

Update (odd).

- $Z_{2k+1}(x) = d_{2k+1}(x) Z_{2k-1}(x) + Z_{2k-1}(x) \times [h(x_{2k}) h(x_{2k+1})] J [h(x_{2k}) h(x_{2k+1})]^{\mathrm{T}} Z_{2k-1}(x)$.

Loop.

- If $k = \lfloor n/2 \rfloor$ then goto [End].
- $k = k + 1$; goto [Generation (even)].

End.

- Return $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$.

Remark 2. The data required by the initialization phase of the algorithm are simply determined as follows. Define $D(x) \doteq \mathrm{diag}(1, x, x^2, \ldots, x^{n-1})$, $\beta \doteq (n/2)(n-1)$, then we have that $Q_0^{(\mathrm{e})}(x) = D(x)Q_0^{(\mathrm{e})}(1)D(x)$, and $\mathrm{Adj}\, D(x) = \mathrm{diag}(x^\beta, x^{\beta-1}, \ldots, x^{\beta-n+1})$, $\mathrm{Adj}\, Q_0^{(\mathrm{e})}(x) = \mathrm{Adj}\, D(x)\mathrm{Adj}\, Q_0^{(\mathrm{e})}(1)\mathrm{Adj}\, D(x)$. Similarly, we get $\mathrm{Adj}\, Q_1^{(\mathrm{e})}(x) = \mathrm{Adj}\, \mathrm{diag}(D(x), x^\beta, x^\beta)\mathrm{Adj}\, Q_1^{(\mathrm{e})}(1)\mathrm{Adj}\, \mathrm{diag}(D(x), x^\beta, x^\beta)$. Therefore, for the even case, we have

$$p_0(x) = x^\beta \sqrt{|Q_0^{(\mathrm{e})}(1)|}, \quad p_1(x) = x^\beta \sqrt{|Q_1^{(\mathrm{e})}(1)|}.$$

We proceed in an analogous way for the odd case, writing $\mathrm{Adj}\, Q_0^{(\mathrm{o})}(x) = \mathrm{Adj}\, \mathrm{diag}(D(x), x^\beta)\mathrm{Adj}\, Q_0^{(\mathrm{o})}(1)\mathrm{Adj}\, \mathrm{diag}(D(x), x^\beta)$, $\mathrm{Adj}\, Q_1^{(\mathrm{o})}(x) = \mathrm{Adj}\, \mathrm{diag}(D(x), x^\beta)\mathrm{Adj}\, Q_1^{(\mathrm{o})}(1)\mathrm{Adj}\, \mathrm{diag}(D(x), x^\beta)$. Therefore, for the odd case, we have

$$p_0(x) = x^\beta \sqrt{|Q_0^{(\mathrm{o})}(1)|}, \quad p_1(x) = x^\beta \sqrt{|Q_1^{(\mathrm{o})}(1)|}.$$

Remark 3. The sign uncertainty on the conditional densities is resolved in the algorithm imposing that the $f_\sigma^{(i)}$'s are positive on their domains.

Notice also that the only operations required for the construction of the conditional densities are simple matrix additions and multiplications. No inversion or computation of determinants of polynomial matrices are required. The generation of each σ_i according to the resulting univariate polynomial density may be performed very efficiently using, for instance, the methods described in Devroye (1986). The computational cost required to generate one sample of Σ is basically n times the cost required to generate one σ_i.

3. A probabilistic framework for uncertain systems

In recent years, we witnessed a widespread use of algorithms based on uncertainty randomization, for the analysis and synthesis of robust control systems, see e.g. Ray and Stengel (1993); Tempo and Dabbene (1999); Vidyasagar and Blondel (2001) and Vidyasagar (2001). The main idea behind randomized methods is to associate a probability distribution to the uncertainty set, and to assess system performance in terms of empirical probability.

We briefly recall below the basic randomized algorithms for estimating empirical probability and the expected value, and discuss various examples of application in the following sections.

Let Δ be a random variable with pdf $f_\Delta(\Delta)$ over the set $\mathbf{\Delta}_\rho$ and let $g(\Delta)$ be a (Lebesgue) measurable function of Δ. The expected value of $g(\Delta)$ is denoted as $E(g(\Delta))$. An empirical estimate $\hat{E}_N$ of $E(g(\Delta))$ is given by

$$\hat{E}_N = \frac{1}{N}\sum_{i=1}^{N} g(\Delta^i),$$

where Δ^i, $i = 1, \ldots, N$ are i.i.d. samples generated according to the pdf $f_\Delta(\Delta)$. The estimate $\hat{E}_N$ is usually referred to as the *empirical mean*. Given *accuracy* $\varepsilon \in (0,1)$ and *confidence* $\delta \in (0,1)$, if

$$N \geqslant \frac{\log \frac{2}{\delta}}{2\varepsilon^2}$$

samples are drawn (Chernoff, 1952), then the empirical mean is close to the actual mean in probability, i.e. $\mathrm{Prob}\{|E_\Delta(g(\Delta)) - \hat{E}_N| \leqslant \varepsilon\} \geqslant 1 - \delta$.

It should be remarked that the sample size N given by the Chernoff bound is independent of the size of $\mathbf{\Delta}_\rho$ and the pdf $f_\Delta(\Delta)$. If the costs associated with the generation of each sample Δ^i and the evaluation of $g(\Delta^i)$ for fixed Δ^i are both polynomial-time, then the estimation of the empirical mean can be performed in polynomial-time. For further details on randomized algorithms, see for instance Tempo and Dabbene (1999) and Vidyasagar (1997). Similarly, the problem of estimating the probability $p_{\mathscr{A}}$ that Δ belongs to a set $\mathscr{A} \subseteq \mathbf{\Delta}_\rho$ is reduced to the computation of the empirical mean, taking as $g(\Delta)$ the indicator function of $\mathscr{A}$.

The next sections present examples of application of the randomized approach to a selection of problems arising in systems and control. In particular, we consider the computation of the solution for uncertain least-squares problems, the probabilistic counterpart of the real structured stability radius, and the assessment of approximate feasibility of uncertain linear matrix inequalities (LMI). An application to reduced order controller design in an H_∞ framework may be found in Calafiore et al. (2000b). Also, a probabilistic approach to the stability analysis of families of parametric polynomials is presented in Polyak and Shcherbakov (2000).

3.1. Uncertain least-squares problems

In this section, we consider the problem of determining approximate solutions to the system of linear equations

$$A(\Delta)x = b(\Delta),$$

where $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and A and b are generic functions (affine or not) of the uncertainty $\Delta \in \mathbf{\Delta}_1$ in form (1). The solution of this problem in a deterministic worst-case setting

is studied in Calafiore and El Ghaoui (2001) and El Ghaoui and Lebret (1997); the worst-case solution is computed as

$$\hat{x}_{\mathrm{RLS}} = \arg\min_{x} \max_{\Delta} \|A(\Delta)x - b(\Delta)\|^2. \tag{19}$$

Here, we assume instead that Δ is a random matrix with given radially symmetric probability distribution over $\mathbf{\Delta}_1$, and seek a solution $\hat{x}$ such that

$$\hat{x} = \arg\min_{x} E\{\|A(\Delta)x - b(\Delta)\|^2\}. \tag{20}$$

To the authors knowledge, the above problem has in general no analytical solution. In a randomized approach, we generate N samples Δ^i of the uncertainty, and look for a solution $\hat{x}_N$ such that the empirical mean is minimized

$$\hat{x}_N = \arg\min_{x} \hat{E}_N(x),$$

where

$$\hat{E}_N(x) \doteq \frac{1}{N}\sum_{i=1}^{N} \|A(\Delta^i)x - b(\Delta^i)\|^2.$$

The solution of the above problem is of course still in the form of an LS problem, and may be efficiently computed recursively. Assuming w.l.o.g. that $A(\Delta^1)$ is full-rank, we have

$$\hat{x}_{k+1} = \hat{x}_k + R_{k+1}^{-1} A^{\mathrm{T}}(\Delta^{k+1})(b(\Delta^{k+1}) - A(\Delta^{k+1})\hat{x}_k),$$

where $R_{k+1} = R_k + A^{\mathrm{T}}(\Delta^{k+1})A(\Delta^{k+1})$, and the recursion for $k = 1,\ldots,N$ is started with $R_0 = 0, \hat{x}_0 = 0$. As a numerical example, we considered the following data, adapted from El Ghaoui and Lebret (1997),

$$A = \begin{bmatrix} 3 & 1 & 4 \\ 0 & 1 & 1 \\ -2 & 5 & 3 \\ 1 & 4 & 5.2 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 3 \end{bmatrix},$$

where $A(\Delta) = A + \Delta$, with $\bar{\sigma}(\Delta) \leqslant 1$. Taking $N = 10,000$ uniform samples of Δ, we obtained the solution $\hat{x}_N^{\mathrm{T}} = [-0.1768\ 0.1009\ 0.3448]$. The standard LS estimate is $\hat{x}_{\mathrm{LS}} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b = [-10.0\ -9.7285\ 9.9834]^{\mathrm{T}}$, and the robust estimate introduced in El Ghaoui and Lebret (1997) is $\hat{x}_{\mathrm{RLS}} = [-0.0312\ 0.2073\ 0.2055]^{\mathrm{T}}$. To compare the results, we computed the estimation residuals $r_N^i \doteq \|A(\Delta^i)\hat{x}_N - b\|$, $r_{\mathrm{LS}}^i \doteq \|A(\Delta^i)\hat{x}_{\mathrm{LS}} - b\|$, $r_{\mathrm{RLS}}^i \doteq \|A(\Delta^i)\hat{x}_{\mathrm{RLS}} - b\|$, for a large number of uncertainty samples Δ^i, and obtained the statistics reported in Table 2. Also, the worst-case (with respect to the uncertainty) residuals, defined as $r_*^{\mathrm{wc}} \doteq \max_{\Delta}\|A(\Delta)x_* - b(\Delta)\|$, result in $r_N^{\mathrm{wc}} = 2.6532, r_{\mathrm{LS}}^{\mathrm{wc}} = 18.9394, r_{\mathrm{RLS}}^{\mathrm{wc}} = 2.572$.

Table 2
Statistics for comparison of residuals: average, peak, and regularity

	Average	Peak	Sample covariance
r_N	2.2650	2.6344	0.0198
r_{LS}	11.8962	18.5164	9.409
r_{RLS}	2.2848	2.5515	0.0107

Problem (20) may therefore be efficiently solved using the proposed randomized approach. The resulting solution has, at least on the data used in the example, a performance which is very close to that of the deterministic worst-case counterpart (19). It should be also remarked that (19) may be solved exactly only in some special cases (see El Ghaoui & Lebret, 1997), while the randomized approach presents no computational complexity issues, and works equally well when the data depends in a non-linear way on the uncertainty, and when the uncertainty has a more complicated block structure.

3.2. The probabilistic structured real stability radius

In this section, we apply the randomized approach to the computation of the probabilistic structured real stability radius of a matrix, see Calafiore, Dabbene, and Tempo (1999b) and Qiu et al. (1995). Given a Hurwitz stable matrix $A \in \mathbb{R}^{p,p}$ and matrices $B \in \mathbb{R}^{p,n}$, $C \in \mathbb{R}^{m,p}$, and assuming $\Delta \sim U[\mathbf{\Delta}_\rho]$, we study the probability of stability $p(\rho) \doteq \mathrm{Prob}\{A + B\Delta C \text{ is stable}\}$. For given $p^* \in [0,1]$, the *probabilistic real stability radius* is defined as $\rho_{\mathbb{R}}(p^*) \doteq \sup\{\rho : p(\rho) \geqslant p^*\}$. In words, given a probability level p^*, the probabilistic real stability radius $\rho_{\mathbb{R}}(p^*)$ gives the maximum "size" of the structured perturbation Δ, measured according to the spectral norm, so that the probability that $A + B\Delta C$ is stable is at least p^*.

The evaluation of $p(\rho)$ can be performed by means of randomized techniques. In particular, given $B = C = I$, and

$$A = \begin{bmatrix} -0.9319 & 0.9633 & 1.1021 & 2.8166 & -1.5852 & -1.3271 \\ -3.5667 & 1.4700 & 2.3962 & 5.2311 & -2.8212 & -4.2641 \\ 1.4202 & -1.1677 & -1.6874 & -3.3362 & 1.0364 & 2.8705 \\ -0.1946 & 0.6813 & 0.0580 & 0.4244 & -0.2107 & -0.6973 \\ 1.2169 & -0.3964 & -0.8681 & -1.9139 & 0.1026 & 0.7190 \\ -2.8445 & 2.0764 & 1.4435 & 4.1812 & -1.8238 & -2.9809 \end{bmatrix},$$

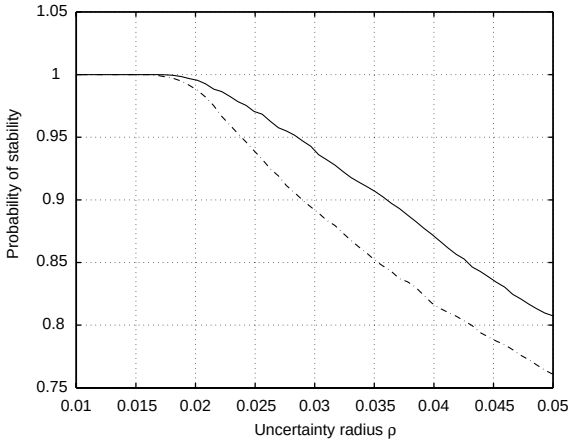

Fig. 1. Estimated probability of stability. The solid line shows the results obtained with a perturbation structure of three 2×2 blocks; the dotted line shows the results obtained with a perturbation structure of a 4×4 and a 2×2 blocks.

we considered two different perturbation structures, one with Δ composed of three 2×2 full real blocks, and one with Δ composed of a 4×4 and a 2×2 full real blocks. Fig. 1 shows the degradation of the empirical probability of stability, as the perturbation radius ρ varies between 0.01 and 0.05.

3.3. Approximate feasibility of uncertain LMIs

An uncertain LMI constraint is a convex constraint in the vector variable $\xi \in \mathbb{R}^m$ of the form

$$F(\xi, \Delta) \prec 0, \quad \Delta \in \mathbf{\Delta}_\rho, \tag{21}$$

where '$\prec$' means "negative definite", and $F(\xi,\Delta)=F_0(\Delta)+\sum_{i=1}^m \xi_i F_i(\Delta)$, with $F_i = F_i^{\mathrm{T}}$. A given vector $\hat{\xi}$ is said to be robustly feasible for (21) if it satisfies (21) for all $\Delta \in \mathbf{\Delta}_\rho$. A large number of problems arising in robust control may be cast as feasibility problems involving uncertain LMIs of the type above, see (Boyd, El Ghaoui, Feron, & Balakrishnan, 1994; El Ghaoui et al., 1998). A classical example is for instance the assessment of quadratic stability of an interval matrix, Boyd et al. (1994). Robust semidefinite programming (SDP) theory (Ben-Tal et al., 2000; El Ghaoui et al., 1998) develops computable sufficient conditions for robust feasibility of uncertain LMIs.

Given a candidate solution $\hat{\xi}$, we here consider the problem of assessing the probability p of satisfaction of (21). The solution $\hat{\xi}$ will be called a p-approximately feasible solution for (21), see also Calafiore and Polyak (2001). To this end, define the sets

$$\mathbf{\Delta}_{\mathrm{good}} \doteq \{\Delta \in \mathbf{\Delta}_\rho : F(\hat{\xi}, \Delta) \prec 0\},$$

$$\mathbf{\Delta}_{\mathrm{bad}} \doteq \{\Delta \in \mathbf{\Delta}_\rho : \Delta \notin \mathbf{\Delta}_{\mathrm{good}}\}.$$

Assuming uniform density over $\mathbf{\Delta}_\rho$, we have

$$p(\rho) \doteq \mathrm{Prob}\{F(\hat{\xi}, \Delta) \prec 0\} = \mathrm{Vol}\{\mathbf{\Delta}_{\mathrm{good}}\}.$$

We then use the randomization procedure to compute an empirical estimate $\hat{p}_N$ of p. This approach has been used for design and analysis of robust LMIs in Calafiore and Polyak (2001), to which the reader is referred for numerical examples.

4. Conclusions

Deterministic worst-case analysis and synthesis methods for uncertain systems are, by and large, based on a structured description of the uncertainty, which is restricted in a spectral (operator) norm bounded set. To compare consistently deterministic results with the recently emerged probabilistic ones, a technique to efficiently generate uncertainty samples uniformly distributed in the above set turned out to be fundamental. This was the main technical issue discussed in this paper.

The proposed sample generation technique relies on the result of Theorem 1 for the closed form expression of the marginal probability densities of the singular values of uniform matrices, and on an efficient recursive implementation of the conditional method for their actual generation.

The use of the proposed framework has then been illustrated, presenting several applications to the solution of hard problems arising in the systems and control area.

Acknowledgements

This paper has benefited from useful discussions and valuable inputs from several people. The authors express gratitude to Laurent El Ghaoui, Augusto Ferrante and Fulvio Ricci. A special thanks goes to Roberto Tempo for its precious help in revising the manuscript, and to an anonymous reviewer for pointing out the interesting set inclusion used in Section 2.

Appendix A. Proof of Lemma 1

Consider (4), and introduce the change of variables $x_k = \sigma_k^2$; $k = 1,\ldots,n$. Then the Jacobian of the transformation from x to σ is $1/(2^n \prod_k \sqrt{x_k})$ (see for instance, Devroye, 1986 for the rule of change of variables in probability density functions), and the pdf in the new variables is expressed as

$$f_x(x_1,\ldots,x_n) = \frac{K_{\mathbb{R}}}{2^n} \prod_{1 \leqslant j < k \leqslant n} (x_j - x_k) \prod_{k=1}^n x_k^v, \tag{A.1}$$

where $1 > x_1 > x_2 > \cdots > x_n > 0$, and $v \doteq (m-n-1)/2$. Notice that f_x can be written in terms of the determinant of a Vandermonde matrix $\mathscr{V}_n$

$$f_x(x_1,\ldots,x_n) = \frac{K_{\mathbb{R}}}{2^n} |\mathscr{V}_n(x_1,\ldots,x_n)| \prod_{k=1}^n x_k^v. \tag{A.2}$$

The marginal densities of x can be then written as

$$f_x^{(i)}(x_1,\dots,x_i) = \frac{K_{\mathbb{R}}}{2^n}\,\Upsilon_i(x_1,\dots,x_i)\prod_{k=1}^{i} x_k^v, \tag{A.3}$$

where $\Upsilon_i \doteq \int_{\mathscr{D}_i} |\mathscr{V}_n|\,\mathrm{d}\mu(x_n)\cdots\mathrm{d}\mu(x_{i+1})$, and $\mathrm{d}\mu(x_k) \doteq x_k^v\,\mathrm{d}x_k$, $\mathscr{D}_i \doteq \{0 < x_n < \cdots < x_i\}$. Statement (8) can be directly obtained from (A.3), applying again the change of variable rule. □

Appendix B. Proof of Lemma 2

For $k = 1,\dots,q/2$, define $\hat{Z}_{2k} \in \mathscr{S}^q$ as

$$\hat{Z}_{2k} = \begin{bmatrix} I_q & 0_{q,2k} \end{bmatrix} Q_{2k}^{-1} \begin{bmatrix} I_q \\ 0_{2k,q} \end{bmatrix}. \tag{B.1}$$

Define $\hat{D}_{2k} \in \mathscr{S}^2$ as $\hat{D}_{2k} = \bar{H}_{2k}^{\mathrm{T}} Q_{2(k-1)}^{-1} \bar{H}_{2k}$, then, in view of (13) and (B.1), we have that

$$\hat{D}_{2k} = H_{2k}^{\mathrm{T}} \hat{Z}_{2(k-1)} H_{2k} = \begin{bmatrix} 0 & \hat{d}(x) \\ -\hat{d}(x) & 0 \end{bmatrix},$$

where $\hat{d}_{2k}(x) = h_{2k-1}^{\mathrm{T}} \hat{Z}_{2(k-1)} h_{2k}$ is a scalar function of x.

We are interested in computing $p_{2k}(x) = \sqrt{|Q_{2k}(x)|}$. Consider then (14) and apply the Schur rule for determinants, obtaining $|Q_{2k}(x)| = |Q_{2(k-1)}| \cdot |H_{2k}^{\mathrm{T}} Q_{2(k-1)}^{-1} H_{2k}| = |Q_{2(k-1)}| \cdot |\hat{D}_{2k}|$. Then $p_{2k}^2(x) = p_{2(k-1)}^2(x)\hat{d}_{2k}^2(x)$. Notice that to compute $\hat{d}_{2k}$ we need $\hat{Z}_{2(k-1)}$, which can be computed recursively. To this aim, we first compute the inverse of Q_{2k}, using the block matrix inversion rule. Setting $\Lambda \doteq Q_{2(k-1)}^{-1} \bar{H}_{2k} \hat{D}_{2k}^{-1}$, we have

$$Q_{2k}^{-1} = \left[\begin{array}{c|c} Q_{2(k-1)}^{-1} - \Lambda \bar{H}_{2k}^{\mathrm{T}} Q_{2(k-1)}^{-1} & -\Lambda \\ \hline \Lambda^{\mathrm{T}} & \hat{D}_{2k}^{-1} \end{array}\right].$$

Using (B.1) it is straightforward to obtain the recursion

$$\hat{Z}_{2k} = \hat{Z}_{2(k-1)} + \frac{1}{\hat{d}_{2k}}\,\hat{Z}_{2(k-1)} H_{2k} J H_{2k}^{\mathrm{T}} \hat{Z}_{2(k-1)}. \tag{B.2}$$

Since we are interested in a recursion involving polynomial matrices, we need to normalize the quantities $\hat{d}_{2k}$ and $\hat{Z}_{2k}$, so to eliminate denominators. It can be shown that the choice $d_{2k} \doteq \hat{d}_{2k}\, p_0^2 \prod_{\ell=1}^{k} d_{2\ell}$; $Z_{2k} \doteq \hat{Z}_{2k}\, p_0^2 \prod_{\ell=1}^{k} d_{2\ell}$, corresponding to the normalization $Z_0 \doteq |Q_0| Q_0 = \mathrm{Adj}\, Q_0$, leads to the desired polynomial recursion given in (17) and (18). □

Appendix C. Technical Preliminaries and Proof of Theorem 2

First, we recall that the determinant of a matrix $X \in \mathscr{S}^n$ is zero if n is odd, and is a perfect square in the entries of X if n is even. The notation $\tilde{X}_{j_1,\dots,j_p}$ denotes the matrix obtained removing the rows and columns of indices $j_1,\dots,j_p$ from matrix X. The notation $\tilde{X}_{i_1,\dots,i_p;j_1,\dots,j_p}$ denotes the matrix obtained removing the rows of indices $i_1,\dots,i_p$, and columns of indices $j_1,\dots,j_p$ from matrix X.

C.1. Preliminaries

Pfaffians: Let $S \in \mathscr{S}^n$, then its *Pfaffian* is defined as the following polynomial in the entries s_{ij} of S

$$\mathrm{Pf}(S) \doteq \frac{1}{2^m m!} \sum_{j_1,\dots,j_n=1}^{n} E(j_1,\dots,j_n) s_{j_1 j_2} s_{j_3 j_4} \cdots s_{j_{2m-1} j_{2m}},$$

where $m = \lfloor n/2 \rfloor$, and $E(x_1,\dots,x_n)$ is an alternant function called *signature function*, defined as $E(x_1,\dots,x_n) \doteq \prod_{1 \leqslant i < j \leqslant n} \mathrm{sign}(x_j - x_i)$. In particular, $E(x_1,\dots,x_n) = 0$ if any two of the x_i's are equal. For further details on the definition and properties of Pfaffians, the reader is referred to Prasolov (1994) and Weyl (1946). A fundamental property of the Pfaffian is that, for n even $\mathrm{Pf}^2(S) = \det(S)$.

The following result on bordered Pfaffians will be useful in the sequel, and may be found in Vein and Dale (1999, Chap. 4).

Lemma C.1. *Let $S \in \mathscr{S}^n$ with n odd, consider the bordered matrix*

$$S^{(1)} = \left[\begin{array}{c|c} S & a_1 \\ \hline -a_1^{\mathrm{T}} & 0 \end{array}\right],$$

where $a_1^{\mathrm{T}} = [a_{11} \cdots a_{n1}]$, then the Pfaffian of $S^{(1)}$ may be expressed as $\mathrm{Pf}(S^{(1)}) = \sum_{j=1}^{n} (-1)^{j+1} a_{j1}\,\mathrm{Pf}(\tilde{S}_j)$.

We next report an extension of the previous result, which may be easily proved by induction.

Lemma C.2. *Let $S \in \mathscr{S}^n$ and consider the bordered matrix*

$$S^{(p)} = \left[\begin{array}{c|c} S & A \\ \hline -A^{\mathrm{T}} & 0 \end{array}\right], \quad A = \begin{bmatrix} a_{11} & \cdots & a_{1p} \\ \vdots & \cdots & \vdots \\ a_{n1} & \cdots & a_{np} \end{bmatrix},$$

where $n + p$ is even. Defining $\ell(A; j_1,\dots,j_p) \doteq (-1)^{j_1+\cdots+j_p+p} E(j_1,\cdots,j_p) a_{j_1,1} a_{j_2,2} \cdots a_{j_p,p}$, the Pfaffian of $S^{(p)}$ may be expressed as

$$\mathrm{Pf}(S^{(p)}) = \sum_{j_1,\dots,j_p=1}^{n} \ell(A; j_1,\dots,j_p)\,\mathrm{Pf}(\tilde{S}_{j_1,\dots,j_p}).$$

C.1.1. A multiple integral of a determinant

We here make use of the following result regarding a multiple integral of a special type of determinant.

Theorem C.3 (de Bruijn, 1955). *Consider the integral*

$$I(\theta) = \int_0^\theta \cdots \int_0^\theta |[\phi(z_1)\ \psi(z_1)\ \cdots\ \phi(z_p)\ \psi(z_p)]| \times \mathrm{d}\mu(z_1)\cdots\mathrm{d}\mu(z_p),$$

where $\phi(\zeta), \psi(\zeta)$ are arbitrary n-dimensional $(n = 2p)$ vector real functions of the real variable ζ, integrable over $[0,\theta]$, and $\mu(\zeta)$ is a suitable measure. Defining the matrix $S(\theta) \in \mathscr{S}^n$, whose (i,j)th element, $i, j = 1,\ldots,n$, is given by

$$[S(\theta)]_{i,j} \doteq \int_0^\theta [\phi_i(\zeta)\psi_j(\zeta) - \psi_i(\zeta)\phi_j(\zeta)]\,\mathrm{d}\mu(\zeta), \qquad \text{(C.1)}$$

we have that $I(\theta) = p!\,\mathrm{Pf}(S(\theta))$.

Below, we provide an extension of de Bruijn's theorem that plays a key role for our subsequent developments.

Theorem C.4. *Consider the integral*

$$I(\theta) = \int_0^\theta \cdots \int_0^\theta |[A\ \ \phi(z_1)\ \psi(z_1)\ \cdots\ \phi(z_p)\ \psi(z_p)]| \times \mathrm{d}\mu(z_1)\cdots\mathrm{d}\mu(z_p), \qquad \text{(C.2)}$$

where $A \in \mathbb{R}^{n,r}$, $2p + r = n$, and $\phi(\zeta), \psi(\zeta)$ are arbitrary n-dimensional vector real functions of the real variable z, integrable over $[0,\theta]$, and $\mu(\zeta)$ is a suitable measure. Define the matrix $S^{(p)}(\theta) \in \mathscr{S}^{n+p}$ as

$$S^{(p)}(\theta) \doteq \left[\begin{array}{c|c} S(\theta) & A \\ \hline -A^{\mathrm{T}} & 0 \end{array}\right],$$

where $S(\theta) \in \mathscr{S}^n$ is defined as in (C.1). *Then*

$$I(\theta) = p!\,\mathrm{Pf}(S^{(p)}(\theta)). \qquad \text{(C.3)}$$

Proof. Define $\Omega = \Omega(z_1,\ldots,z_p) \doteq [A\ \phi(z_1)\ \psi(z_1)\ \cdots\ \phi(z_p)\ \psi(z_p)]$. We first expand the determinant in (C.2) with respect to the columns of A, using the Laplace expansion (Vein & Dale, 1999),

$$|\Omega| = \sum_{j_1,\ldots,j_p=1}^{n} \ell(A; j_1,\ldots,j_p)|\tilde{\Omega}_{j_1,\ldots,j_p;1,\ldots,p}|. \qquad \text{(C.4)}$$

Integrating (C.4) we obtain

$$I(\theta) = \sum_{j_1,\ldots,j_p=1}^{n} \ell(A; j_1,\ldots,j_p) \int_0^\theta \cdots \int_0^\theta |\tilde{\Omega}_{j_1,\ldots,j_p;1,\ldots,p}| \times \mathrm{d}\mu(z_1)\cdots\mathrm{d}\mu(z_p).$$

The integrals appearing in the above expression may be computed using Theorem 3, obtaining

$$\int_0^\theta \cdots \int_0^\theta |\tilde{\Omega}_{j_1,\ldots,j_p;1,\ldots,p}|\,\mathrm{d}\mu(z_1)\cdots\mathrm{d}\mu(z_p) = p!\,\mathrm{Pf}(Z(\theta)),$$

where $Z(\theta)$ also depends on the indices $(j_1,\ldots,j_p)$, and may be recognized to be a principal submatrix of the skew-symmetric matrix $S(\theta)$ defined in (C.1), that is $Z(\theta) \equiv \tilde{S}_{j_1,\ldots,j_p}(\theta)$. Integral (C.2) can therefore be written as

$$I(\theta) = p! \sum_{j_1,\ldots,j_p=1}^{n} \ell(A; j_1,\ldots,j_p)\,\mathrm{Pf}(\tilde{S}_{j_1,\ldots,j_p}(\theta)). \qquad \text{(C.5)}$$

The proof then follows applying Lemma C.2 to (C.5). □

C.2. Proof of Theorem 2

To prove the theorem we need to consider two separate cases, depending on whether $n - i$ is even or odd.

Case $n - i$ even: Notice first that in the integral (9), each column of $\mathscr{V}_n(x_1,\ldots,x_n)$ is function of only one variable. We can therefore integrate using the method of integration over alternate variables (Mehta, 1991). First, we integrate $x_n, x_{n-2},\ldots,x_{i+2}$ over their respective domains: let $F(z) \doteq \int_0^z \mathscr{X}(\xi)\,\mathrm{d}\mu(\xi)$, then

$$\Upsilon_i = \int_{\mathscr{D}_i} |[\mathscr{V}_i\ \mathscr{X}(x_{i+1})\ F(x_{i+1}) - F(x_{i+3})\ \mathscr{X}(x_{i+3})\ F(x_{i+3}) - F(x_{i+5})\ \cdots\ \mathscr{X}(x_{n-1})\ F(x_{n-1})]| \times \mathrm{d}\mu(x_{n-1})\cdots\mathrm{d}\mu(x_{i+5})\,\mathrm{d}\mu(x_{i+3})\,\mathrm{d}\mu(x_{i+1}).$$

The addition of one column to another column does not change the determinant of a matrix, therefore

$$\Upsilon_i = \int_{\mathscr{D}_i} |[\mathscr{V}_i\ \mathscr{X}(x_{i+1})\ F(x_{i+1})\ \cdots\ \mathscr{X}(x_{n-1})\ F(x_{n-1})]| \times \mathrm{d}\mu(x_{n-1})\,\mathrm{d}\mu(x_{n-3})\cdots\mathrm{d}\mu(x_{i+1}). \qquad \text{(C.6)}$$

Notice now that the integrand in (C.6) is symmetric in the remaining variables[1] $x_{i+1}, x_{i+3},\ldots,x_{n-1}$, therefore one can integrate over them independently (see Mehta, 1991) and divide the result by $((n-i)/2)!$, obtaining

$$\Upsilon_i = \left(\frac{n-i}{2}!\right)^{-1} \int_0^{x_i} \cdots \int_0^{x_i} |[\mathscr{V}_i\ \mathscr{X}(x_{i+1})\ F(x_{i+1})\ \cdots\ \mathscr{X}(x_{n-1})\ F(x_{n-1})]|\,\mathrm{d}\mu(x_{n-1})\,\mathrm{d}\mu(x_{n-3})\cdots\mathrm{d}\mu(x_{i+1}).$$

We can now apply the result of Theorem C.3 to the above integral, obtaining

$$\Upsilon_i(x_1,\ldots,x_i) = \mathrm{Pf}\left[\begin{array}{c|c} S(x_i) & \mathscr{V}_i \\ \hline -\mathscr{V}_i^{\mathrm{T}} & 0 \end{array}\right], \qquad \text{(C.7)}$$

[1] The integrand is symmetric in the variables, since interchanging x_i and x_j amounts to the interchanging of two *pairs* of columns.

where $S(x_i) \in \mathscr{S}^n$ is defined as in (C.1), with $\phi_k(x) \doteq \mathscr{X}_k(x) = x^{k-1}$, and $\psi_k(x) \doteq F_k(x) = \int_0^x \xi^{k-1}\xi^v \,\mathrm{d}\xi = (x^{k+v}/(k+v))$, for $k = 1,\ldots,n$. Therefore,

$$S_{rj}(x_i) = \frac{r-j}{(r+v)(j+v)(r+j+2v)} x_i^{r+j+2v}.$$

Define now $\bar{S}(x_i)$ as in (12), such that $S(x_i) = x_i^{2(v+1)}\bar{S}(x_i)$. Then, recalling that $\mathscr{V}_i = [\mathscr{V}_{i-1}\ \mathscr{X}_i]$, and that the exchange of one row and one column does not affect the determinant, factoring out the term $x_i^{2(v+1)}$ from the Pfaffian, we rewrite (C.7) as

$$\Upsilon_i = x_i^{(v+1)(n-i)} \operatorname{Pf} \left[\begin{array}{cc|c} \bar{S}(x_i) & \mathscr{X}(x_i) & \mathscr{V}_{i-1} \\ -\mathscr{X}^{\mathrm{T}}(x_i) & 0 & 0 \\ \hline -\mathscr{V}_{i-1}^{\mathrm{T}} & 0 & 0 \end{array}\right], \qquad \text{(C.8)}$$

from which follows the statement of (11).

Case $n-i$ odd. We now first integrate $x_n, x_{n-2}, \ldots, x_{i+1}$ over their respective domains, obtaining

$$\begin{aligned}\Upsilon_i = \int_{\mathscr{D}_i} |[&\mathscr{V}_i \ F(x_i) - F(x_{i+2}) \ \mathscr{X}(x_{i+2}) \\ &F(x_{i+2}) - F(x_{i+4}) \ \cdots \ \mathscr{X}(x_{n-1}) \ F(x_{n-1})]| \\ &\times \mathrm{d}\mu(x_{n-1}) \cdots \mathrm{d}\mu(x_{i+4}) \,\mathrm{d}\mu(x_{i+2}).\end{aligned}$$

Again, performing elementary operations on the columns, we obtain

$$\begin{aligned}\Upsilon_i = \int_{\mathscr{D}_i} |[&\mathscr{V}_i \ F(x_i) \ \mathscr{X}(x_{i+2}) \ F(x_{i+2}) \ \cdots \ \mathscr{X}(x_{n-1}) \\ &F(x_{n-1})]| \,\mathrm{d}\mu(x_{n-1}) \cdots \mathrm{d}\mu(x_{i+4}) \,\mathrm{d}\mu(x_{i+2}).\end{aligned}$$

The integrand above is symmetric in the remaining variables $x_{i+2}, x_{i+4} \ldots, x_{n-1}$, therefore integrating over them independently, and dividing the result by $((n-i-1)/2)!$ (Mehta, 1991), we obtain

$$\begin{aligned}\Upsilon_i = \left(\frac{n-i-1}{2}!\right)^{-1} \int_0^{x_i} \cdots \int_0^{x_i} |[&\mathscr{V}_i \ F(x_i) \\ &\mathscr{X}(x_{i+2}) F(x_{i+2}) \cdots \ \mathscr{X}(x_{n-1}) \ F(x_{n-1})]| \\ &\times \mathrm{d}\mu(x_{n-1}) \cdots \mathrm{d}\mu(x_{i+4}) \,\mathrm{d}\mu(x_{i+2}).\end{aligned}$$

We can now apply the result of Theorem 3 to the above integral, obtaining

$$\Upsilon_i(x_1,\ldots,x_i) = \operatorname{Pf} \left[\begin{array}{c|cc} S(x_i) & \mathscr{V}_i & F(x_i) \\ \hline -\mathscr{V}_i^{\mathrm{T}} & & \\ & 0 & \\ -F^{\mathrm{T}}(x_i) & & \end{array}\right], \qquad \text{(C.9)}$$

where $S(x_i) \in \mathscr{S}^n$ is defined as in (1). Defining now $\bar{S}(x_i)$ and $\bar{F}(x_i)$ as in (12), the result in (11) is then obtained following the same reasoning as in the even case. □

References

Anderson, T. W. (1984). *An introduction to multivariate statistical analysis*. New York: Wiley.

Bai, E. W., Tempo, R., & Fu, M. (1998). Worst case properties of the uniform distribution and randomized algorithms for robustness analysis. *Mathematics of Control, Signals and Systems*, *11*, 183–196.

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

Ben-Tal, A., El Ghaoui, L., & Nemirovskii, A. (2000). Robust Semidefinite programming. In R. Saigal, L. Vandenberghe, & H. Wolkowicz (Eds.), *Handbook of semidefinite programming*. Waterloo: Kluwer Academic Publishers.

Blondel, V. D., & Tsitsiklis, J. N. (2000). A survey of computational complexity results in systems and control. *Automatica*, *36*, 1249–1274.

Boyd, S. P., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in systems and control theory*. Philadelphia, PA: SIAM Publications.

Calafiore, G., Dabbene, F., & Tempo, R. (1999a). Radial and uniform distributions in vector and matrix spaces for probabilistic robustness. In D. Miller, & L. Qiu (Eds.), *Topics in control and its applications* (pp. 17–31). London: Springer.

Calafiore, G., Dabbene, F., & Tempo, R. (1999b). The probabilistic real stability radius. *Proceedings of the 14th IFAC world congress*, Vol. G, Beijing, July 1999 (pp. 395–400).

Calafiore, G., Dabbene, F., & Tempo, R. (2000a). Randomized algorithms for probabilistic robustness with real and complex structured uncertainty. *IEEE Transactions on Automatic Control*, *45*(12), 2218–2235.

Calafiore, G., Dabbene, F., & Tempo, R. (2000b). Randomized algorithms for reduced order H_∞ controller design. *Proceedings of the American control conference*, Vol. 6, Chicago, June 2000 (pp. 3837–3839).

Calafiore, G., & El Ghaoui, L. (2001). Worst case maximum likelihood estimation in the linear model. *Automatica*, *37*(4), 573–580.

Calafiore, G., & Polyak, B. T. (2001). Stochastic algorithms for exact and approximate feasibility of robust LMIs. *IEEE Transactions on Automatic Control*, *46*(11), 1755–1759.

Chernoff, H. (1952). A measure of asymptotic efficiency for test of hypothesis based on the sum of observations. *Annals of Mathematics and Statistics*, *23*, 493–507.

de Bruijn, N. G. (1955). On some multiple integrals involving determinants. *Journal of the Indian Mathematical Society*, *19*, 127–132.

Devroye, L. (1986). *Non-uniform random variate generation*. New York: Springer.

Devroye, L. (1997). Random variate generation for multivariate unimodal densities. *ACM Transactions on Modelling and Computer Simulations*, *7*, 447–477.

El Ghaoui, L., & Calafiore, G. (2001). Robust filtering for discrete-time systems with bounded noise and parametric uncertainty. *IEEE Transactions on Automatic Control*, *46*(7), 1084–1089.

El Ghaoui, L., & Lebret, H. (1997). Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, *18*(4), 1035–1064.

El Ghaoui, L., Oustry, F., & Lebret, H. (1998). Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, *9*(1), 33–52.

Khargonekar, P. P., & Tikku, A. (1996). Randomized algorithms for robust control analysis have polynomial time complexity. *Proceedings of the 35th IEEE conference on decision and control*, Vol. 3, Kobe, December 1996 (pp. 3470–3475).

Mehta, M. L. (1991). *Random matrices*. Boston: Academic Press.

Nemirovskii, A. (1993). Several NP-hard problems arising in robust stability analysis. *Mathematics of Control, Signals and Systems*, *6*, 99–105.

Polijak, S., & Rohn, J. (1993). Checking robust non-singularity is NP-hard. *Mathematics of Control, Signals and Systems*, *6*, 1–9.

Polyak, B. T., & Shcherbakov, P. S. (2000). Random spherical uncertainty in estimation and robustness. *IEEE Transactions on Automatic Control*, *45*(11), 2145–2150.

Prasolov, V. V. (1994). *Problems and theorems in linear algebra*. Providence, RI: American Mathematical Society.

Qiu, L., Bernhardsson, L. B., Rantzer, A., Davison, E. J., Young, P. M., & Doyle, J. (1995). A formula for computation of the real stability radius. *Automatica*, *6*, 879–890.

Ray, L. R., & Stengel, R. F. (1993). A Monte carlo approach to the analysis of control system robustness. *Automatica*, *29*, 229–236.

Selberg, A. (1944). Bemerkninger om et multiplet integral. *Norsk Matematisk Tidsskrift*, *26*, 71–78.

Stewart, G. W. (1980). The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM Journal on Numerical Analysis*, *17*(3), 403–409.

Tempo, R., Bai, E. W., & Dabbene, F. (1997). Probabilistic robustness analysis: Explicit bounds for the minimum number of samples *Systems & Control Letters*, *30*, 237–242.

Tempo, R., & Dabbene, F. (1999). Probabilistic robustness analysis and design of uncertain systems. In G. Picci, & D. S. Gilliam (Eds.), *Dynamical systems*, *control*, *coding*, *computer vision—New trends*, *interfaces*, *and interplay*, Vol. 25 (pp. 263–282). Boston: Birkhäuser.

Vein, R., & Dale, P. (1999). *Determinants and their applications in mathematical physics*. New York: Springer.

Vidyasagar, M. (1997). *A theory of learning and generalization*. London: Springer.

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

Vidyasagar, M., & Blondel, V. D. (2001). Probabilistic solutions to some NP-hard matrix problems. *Automatica*, *37*(9), 1397–1405.

Weyl, H. (1946). *The classical groups, their invariants and representations*. Princeton, NJ: Princeton University Press.

Xie, L., Soh, Y. C., & de Souza, C. E. (1994). Robust Kalman filtering for uncertain discrete-time systems. *IEEE Transactions on Automatic Control*, *39*(6), 1310–1314.

Zhou, K., Doyle, J. C., & Glover, K. (1996). *Robust and optimal control*. Upper Saddle River: Prentice-Hall.

Zhu, X., Huang, Y., & Doyle, J. (1996). Soft vs. hard bounds in probabilistic robustness analysis. *Proceedings of the 35th IEEE conference on decision and control*. Vol. 3, Kobe, Japan (pp. 3412–3417).

Giuseppe Calafiore was born in Torino, Italy in 1969. He received the "Laurea" degree in Electrical Engineering from Politecnico di Torino in 1993, and the Doctorate degree in Information and System Theory from Politecnico di Torino, in 1997. Since 1998, he is tenured Assistant Professor at Dipartimento di Automatica e Informatica, Politecnico di Torino. Dr. Calafiore held visiting positions at Information Systems Laboratory, Stanford University, in 1995, at Ecole Nationale Supérieure de Techniques Avenceés (ENSTA), Paris, in 1998, and at the University of California at Berkeley, in 1999. His research interests are in the field of analysis, identification and control of uncertain systems, pattern analysis and robotics, convex optimization and randomized algorithms.

Fabrizio Dabbene received the Laurea degree in Electronic Engineering in 1995 and the Ph.D. degree in Systems and Computer Engineering in 1999, both from Politecnico di Torino, Italy. In 1997, he has been Visiting Researcher at the Department of Electrical Engineering, University of Iowa. He is currently holding a tenured research position at the research institute IRITI of the National Research Council (CNR) of Italy Politecnico di Torino. Dr. Dabbene also holds collaboration with Politecnico di Torino where he teaches several courses in Systems and Control. His research interests include robust control and identification of uncertain systems, randomized algorithms and optimization.