MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Polynomial-time algorithms for probabilistic solutions of parameter-dependent linear matrix inequalities

Yasuaki Oishi

METR 2004–23 May 2004

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm

Polynomial-time algorithms for probabilistic solutions of parameter-dependent linear matrix inequalities*

Yasuaki Oishi†

Difficulties and their fundamental resolutions are presented on gradient-based and ellipsoid-based randomized algorithms for solving parameter-dependent linear matrix inequalities. Those algorithms in their original forms have the following difficulties: (i) Appropriate choice of a step-size parameter or an initial ellipsoid is difficult; (ii) Detection of convergence is difficult; (iii) The expected number of necessary iterations is infinite. This paper resolves these difficulties by introducing stopping rules into the algorithms. The resulting algorithms always stop in a bounded number of iterations and this bound is of polynomial order in the problem size. When the algorithms stop, they either give probabilistic solutions with high confidence or detect that there is no deterministic solution in an approximated sense. The algorithms can be adapted for finding an optimal solution of a parameter-dependent linear matrix inequality. Usefulness of the proposed algorithms is illustrated by a numerical example.

Keywords: randomized algorithms, parameter-dependent linear matrix inequalities, stopping rules, probabilistic solutions, computational complexity, optimization.

1. Introduction

A parameter-dependent linear matrix inequality (LMI) is important because many problems on analysis and design of robust and gain-scheduled control systems are formulated in this form. See [2, 25, 1, 8] for example. In the case that the parameter-dependence is nonlinear, we often reduce it to a set of parameter-independent LMIs by gridding a parameter set or by covering the set with a polytope. These conventional approaches, however, have drawbacks: The gridding approach does not guarantee the inequality to be satisfied between the grid points; The covering approach is conservative, that is, even if the original parameter-dependent LMI has a solution, the resulting set of parameter-independent LMIs may not. Recently, Ohara and Sasaki [13] and Scherer [21] proposed new approaches in a special case that parameter-dependence is expressed in a rational function. These approaches construct a sequence of parameter-independent LMIs, which gives an asymptotically exact approximation of the original parameter-dependent LMI. Although these approaches are attractive, their computational complexity is problematic in that the size of the approximating LMI grows exponentially with the parameter dimension [13] or the approximation level [21].

*Released on May 10, 2004

†Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan; phone: +81-3-5841-6921; fax: +81-3-5841-6886; email: oishi@mist.i.u-tokyo.ac.jp

In order to overcome the limitation of the existing approaches, Polyak and Tempo [20] and Calafiore and Polyak [6] proposed a randomized algorithm with further extensions in [19, 10]. This algorithm enables us to get rid of conservatism in solving parameter-dependent LMIs and to treat nonlinear parameter dependence. It is also applicable to the case that a parameter has a probabilistic structure, that is, some parameter values are probable and others are not. This algorithm iterates random sampling of a parameter value and update of a solution candidate by a gradient-based method [20, 6, 19] or by an ellipsoid-based method [10]. It is proved that with probability one in a finite number of iterations a candidate reaches a deterministic solution that satisfies the given LMI for all parameter values.

The first contribution of this paper is to point out that the above randomized algorithm has the following difficulties in its practical applications. First, some assumptions must be satisfied in the choice of a step-size parameter for the gradient-based algorithm and an initial ellipsoid for the ellipsoid-based algorithm, which is not always easy. Second, it is difficult to know when a candidate arrives at a deterministic solution. Finally, the number of iterations necessary to find a deterministic solution is a random number distributed with a heavy tail. In fact, it is shown in [18] that the expectation of this number is infinite in the case of the gradient-based algorithm. We prove in this paper a corresponding result not only on the gradient-based algorithm but also on the ellipsoid-based one in a stronger form where an initial candidate of a solution is chosen deterministically. The second contribution of this paper is to resolve the above difficulties by introducing a stopping rule into the randomized algorithm. The resulting algorithm always stops and the number of iterations is bounded by a polynomial in the problem size. When it stops, it either gives a probabilistic solution with high confidence or detects that there is no deterministic solution in an approximated sense. We also extend this algorithm for finding an optimal solution of a parameter-dependent LMI. Practical usefulness of the proposed algorithm is shown by a numerical example.

While a technique of Fujisaki *et al.* [7] and Oishi and Kimura [18] resolves the third difficulty, it does not the first and the second difficulties. Kanev *et al.* [10] gave some stopping rules without theoretical analysis. Earlier versions of the present work have been presented at SICE conferences [14, 15] and at the CDC2003 [16]. Independently of this work, Calafiore *et al.* [5] presented a closely related work on a stopping rule at the CDC2003. A major difference is that the stopping rule in the present work gives better computational complexity than theirs. The present results on the expected number of iterations and on the extension to optimization seem to be new.

In Section 2, a parameter-dependent LMI is provided. In Section 3, the gradient-based randomized algorithm is presented and its difficulties are pointed out. Section 4 shows that introduction of a stopping rule resolves the problems stated before. Section 5 is devoted to deriving corresponding results on the ellipsoid-based randomized algorithm. Extension to an optimization

problem is considered in Section 6 and a numerical example is provided in Section 7. Section 8 concludes the paper.

Throughout this paper, ln stands for the natural logarithm. For a real number a, the symbol $\lceil a \rceil$ denotes the minimum integer that is larger than or equal to a. The symbol $\mathbb{R}^n$ designates the n-dimensional Euclidean space and vol the volume in $\mathbb{R}^n$. The symbol $\|\cdot\|$ denotes the Euclidean norm. The transpose of a matrix or a vector is expressed by $^{\mathrm{T}}$. The maximum (most positive) eigenvalue of a symmetric matrix A is written as $\overline{\lambda}[A]$. Negative definiteness and negative semidefiniteness of a symmetric matrix A are denoted by $A \prec O$ and $A \preceq O$, respectively.

2. Problem

Let $V(\boldsymbol{x}, \boldsymbol{\theta})$ be a symmetric-matrix-valued function of a variable $\boldsymbol{x} \in \mathbb{R}^n$ and a parameter $\boldsymbol{\theta} \in \Theta$, where Θ is a given subset of $\mathbb{R}^p$.

The function $V(\boldsymbol{x}, \boldsymbol{\theta})$ is assumed to be affine in $\boldsymbol{x}$ and is written as

$$V(\boldsymbol{x}, \boldsymbol{\theta}) = V_0(\boldsymbol{\theta}) + \sum_{i=1}^{n} x_i V_i(\boldsymbol{\theta})$$

with x_i being the ith element of $\boldsymbol{x}$. At this moment, no assumption is made on the set Θ or on the dependence of $V(\boldsymbol{x}, \boldsymbol{\theta})$ on $\boldsymbol{\theta}$. We consider the following problem.

Problem 1. Find $\boldsymbol{x} \in \mathbb{R}^n$ that satisfies $V(\boldsymbol{x}, \boldsymbol{\theta}) \prec O$ for all $\boldsymbol{\theta} \in \Theta$. □

The inequality $V(\boldsymbol{x}, \boldsymbol{\theta}) \prec O$ is called an LMI dependent on the parameter $\boldsymbol{\theta} \in \Theta$. Analysis and design of robust and gain-scheduled control systems are often formulated in this form [2, 25, 1, 8].

The $\boldsymbol{x}$ desired in Problem 1 is referred to as a *deterministic solution*. The set of all deterministic solutions is called the *solution set* $S \subseteq \mathbb{R}^n$. In the succeeding sections, we will consider to obtain an approximate solution rather than a deterministic solution itself. By a *non-strict deterministic solution*, we mean an $\boldsymbol{x} \in \mathbb{R}^n$ that satisfies $V(\boldsymbol{x}, \boldsymbol{\theta}) \preceq O$ for all $\boldsymbol{\theta} \in \Theta$. Note that the strict inequality in Problem 1 is replaced by a non-strict one. Moreover, for a given probability measure P on Θ and a given $0 < \epsilon < 1$, we mean by a *probabilistic solution* an $\boldsymbol{x} \in \mathbb{R}^n$ that satisfies $P\{\boldsymbol{\theta} \in \Theta : V(\boldsymbol{x}, \boldsymbol{\theta}) \prec O\} > 1 - \epsilon$.

Noting that $\overline{\lambda}[V(\boldsymbol{x}, \boldsymbol{\theta})] < 0$ is equivalent to $V(\boldsymbol{x}, \boldsymbol{\theta}) \prec O$, we often use this alternative expression for negative definiteness. It is known that $\overline{\lambda}[V(\boldsymbol{x}, \boldsymbol{\theta})]$ is a convex function in $\boldsymbol{x}$ and its subgradient is given by $[\boldsymbol{v}^{\mathrm{T}} V_1(\boldsymbol{\theta}) \boldsymbol{v} \quad \cdots \quad \boldsymbol{v}^{\mathrm{T}} V_n(\boldsymbol{\theta}) \boldsymbol{v}]^{\mathrm{T}}$, where $\boldsymbol{v}$ is any normalized eigenvector corresponding to the maximum eigenvalue $\overline{\lambda}[V(\boldsymbol{x}, \boldsymbol{\theta})]$. See, *e.g.*, [9, Section D.5.1].

3. Gradient-based algorithm

The gradient-based randomized algorithm proposed by Polyak and Tempo [20] and Calafiore and Polyak [6] can be used for finding a non-strict deterministic solution of Problem 1. In this section, we present this algorithm and discuss its difficulties.

Before executing the algorithm, we need to choose an initial candidate of a solution $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ and a step-size parameter $r > 0$. We make the following assumptions.

Assumption 2. For $\boldsymbol{x}^* \in \mathbb{R}^n$, the probability $P\{\boldsymbol{\theta} \in \Theta : \overline{\lambda}[V(\boldsymbol{x}^*, \boldsymbol{\theta})] > 0\}$ is positive whenever $\overline{\lambda}[V(\boldsymbol{x}^*, \boldsymbol{\theta})] > 0$ holds for some $\boldsymbol{\theta} \in \Theta$. □

Assumption 3. The solution set S includes an open ball of radius r. □

Assumption 2 is reasonable because it is satisfied whenever $V_i(\boldsymbol{\theta})$, $i = 0, 1, \ldots, n$, are continuous in $\boldsymbol{\theta}$ and P has a positive density function. Assumption 3 is not considered reasonable; we will discuss this later.

Algorithm 4.

0. Set $k := 0$.

1. Randomly sample $\boldsymbol{\theta}^{(k)} \in \Theta$ according to the given probability measure P.

2. Check $\lambda^{(k)} := \overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)})]$ for negativity.

   Case 1 The value $\lambda^{(k)}$ is nonnegative.
   Compute a subgradient, say $\boldsymbol{d}^{(k)}$, of $\overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)})]$ as a convex function in $\boldsymbol{x}$. Update the current solution candidate by setting

$$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} - \Big(\frac{\lambda^{(k)}}{\|\boldsymbol{d}^{(k)}\|} + r\Big)\frac{\boldsymbol{d}^{(k)}}{\|\boldsymbol{d}^{(k)}\|}.$$

   Case 2 The value $\lambda^{(k)}$ is negative.
   Keep the current solution candidate by setting $\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)}$.

3. Set $k := k + 1$. Go back to Step 1. □

The original algorithm of [20, 6] computed the subgradient of a different function. We prefer to use the function $\overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)})]$ following [12] for the reason stated in Remark 15. Moreover, the original algorithm of [20, 6] was given in a more general form; in particular, the domain of $\boldsymbol{x}$ can be restricted to a subset of $\mathbb{R}^n$ and an over-relaxation parameter is used. We focus on the simplest case in this paper because generalizations are easy.

The performance of this algorithm is guaranteed by the next proposition, which is a restatement of Theorem 1 of [20] and Theorem 1 of [6]. Notice that the probabilistic behavior of this algorithm is determined by the sequence $\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \ldots$, each term of which is distributed independently and identically according to P. The performance of this algorithm can be analyzed with the probability measure on such sequences, which is derived from P and is denoted by P^∞. See for example [22, Section II.3.4].

Proposition 5. *Under Assumptions 2 and 3, there exists with probability one a finite k such that $\boldsymbol{x}^{(k)}$ produced by Algorithm 4 is a non-strict deterministic solution, where the probability is measured with P^∞.*

For each sequence $\{\boldsymbol{\theta}^{(k)}\}$, there may exist multiple k's having the property of the proposition. The minimum such k is denoted by k_{N}, which stands for the number of iterations necessary to find a non-strict deterministic solution. Note that k_{N} is a random number.

This result has some practical difficulties though it is theoretically attractive. First, making Assumption 3 is not always reasonable. In many cases, it is already difficult to know whether the solution set S is nonempty, not to mention whether it includes an open ball of radius r. If Assumption 3 fails to hold for the chosen value of r, the finite-time convergence is not guaranteed. Second, it is difficult to detect when $\boldsymbol{x}^{(k)}$ arrives at a non-strict deterministic solution. Indeed, in order to detect it, we need to check $V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}) \preceq O$ for all $\boldsymbol{\theta} \in \Theta$, which is impossible in a typical case that Θ consists of infinitely many parameter values. Finally, the number of necessary iterations k_{N} is distributed with a heavy tail. In fact, we will prove below that the expectation of this number is infinite. These difficulties imply that the theoretical result of Proposition 5 has only limited practical significance, which motivates the improvement in the next section.

In the rest of this section, we give a precise statement on the expectation of k_{N}.

Theorem 6. *Suppose that an initial solution candidate $\boldsymbol{x}^{(0)}$ is provided. Under Assumptions 7–10 below, the number of necessary iterations k_{N} in Algorithm 4 has infinite expectation, where the expectation is taken with the measure P^∞.*

A proof of this theorem is found in Appendix A. An intuitive reason for this to hold is as follows. Suppose that a solution candidate $\boldsymbol{x}^{(k)}$ is not a non-strict deterministic solution but is close to the solution set S. Then, while we still need to update $\boldsymbol{x}^{(k)}$, the value $\overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta})]$ is negative for most of $\boldsymbol{\theta} \in \Theta$. Since $\boldsymbol{x}^{(k)}$ is updated only when $\overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)})]$ is nonnegative for a randomly sampled $\boldsymbol{\theta}^{(k)}$, we are likely to wait for many iterations before $\boldsymbol{x}^{(k)}$ is updated. By an explicit evaluation of the probability of update, we can show the theorem.

An earlier version of this theorem has been given in [18]. The present version is stronger in the sense that an initial solution candidate $\boldsymbol{x}^{(0)}$ is chosen in a deterministic way and not in a probabilistic way as in the previous version.

We present technical assumptions required in the theorem, which are considered to be plausible.

Assumption 7. The parameter set Θ is a bounded closed set having a nonempty interior. There exists an open set that includes Θ and has $V_0(\boldsymbol{\theta}), \ldots, V_n(\boldsymbol{\theta})$ continuously differentiable there. The probability measure P has a density function possessing a finite upper bound and a positive lower bound. □

Assumption 8. There exists a finite sequence of parameter values $\{\widehat{\boldsymbol{\theta}}^{(k)}\}_{k=0}^{k^*}$ having the following properties, where $\widehat{\boldsymbol{\theta}}^{(0)}, \ldots, \widehat{\boldsymbol{\theta}}^{(k^*-1)}$ are in the interior of Θ and $\widehat{\boldsymbol{\theta}}^{(k^*)}$ is in Θ. If we choose $\boldsymbol{\theta}^{(0)} = \widehat{\boldsymbol{\theta}}^{(0)}$, $\ldots$, $\boldsymbol{\theta}^{(k^*)} = \widehat{\boldsymbol{\theta}}^{(k^*)}$ in Algorithm 4, we obtain $\boldsymbol{x}^{(0)} = \widehat{\boldsymbol{x}}^{(0)}$, $\ldots$, $\boldsymbol{x}^{(k^*)} = \widehat{\boldsymbol{x}}^{(k^*)}$, with which we have

(a) $\overline{\lambda}[V(\widehat{\boldsymbol{x}}^{(k)}, \widehat{\boldsymbol{\theta}}^{(k)})] > 0 \quad \text{for } k = 0, \ldots, k^* - 1;$

(b) $\overline{\lambda}[V(\widehat{\boldsymbol{x}}^{(k^*)}, \widehat{\boldsymbol{\theta}}^{(k^*)})] = \max_{\boldsymbol{\theta} \in \Theta} \overline{\lambda}[V(\widehat{\boldsymbol{x}}^{(k^*)}, \boldsymbol{\theta})] = 0;$

(c) The maximum eigenvalue $\overline{\lambda}[V(\widehat{\boldsymbol{x}}^{(k)}, \widehat{\boldsymbol{\theta}}^{(k)})]$ is simple for each of $k = 0, \ldots, k^*$;

(d) The gradient of $\overline{\lambda}[V(\boldsymbol{x}, \boldsymbol{\theta})]$ with respect to $\boldsymbol{\theta}$ is nonzero at $(\widehat{\boldsymbol{x}}^{(k^*)}, \widehat{\boldsymbol{\theta}}^{(k^*)})$. □

The assumption (c) guarantees continuous differentiability of $\overline{\lambda}[V(\boldsymbol{x}, \boldsymbol{\theta})]$ in neighborhoods of $(\widehat{\boldsymbol{x}}^{(k)}, \widehat{\boldsymbol{\theta}}^{(k)})$, $k = 0, \ldots, k^*$. This makes the assumption (d) meaningful. Note also that as a consequence of (b) and (d) the point $\widehat{\boldsymbol{\theta}}^{(k^*)}$ has to lie on the boundary of Θ.

Since $\boldsymbol{x}^{(k^*)}$ depends on the choice of $\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(k^*-1)}$, it is possible to regard $\overline{\lambda}[V(\boldsymbol{x}^{(k^*)}, \boldsymbol{\theta}^{(k^*)})]$ as a function of $\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(k^*)}$. This function is continuously differentiable in a neighborhood of $(\widehat{\boldsymbol{\theta}}^{(0)}, \ldots, \widehat{\boldsymbol{\theta}}^{(k^*)})$ by continuous differentiability assumed in Assumption 7, simplicity of the eigenvalue assumed in Assumption 8 (c), and the update rule of Algorithm 4. It is also possible to regard $\max_{\boldsymbol{\theta}^{(k^*)} \in \Theta} \overline{\lambda}[V(\boldsymbol{x}^{(k^*)}, \boldsymbol{\theta}^{(k^*)})]$ as a function of $\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(k^*-1)}$. On this function, the next assumption is made.

Assumption 9. The maximum value, $\max_{\boldsymbol{\theta}^{(k^*)} \in \Theta} \overline{\lambda}[V(\boldsymbol{x}^{(k^*)}, \boldsymbol{\theta}^{(k^*)})]$, is attained at a unique $\boldsymbol{\theta}^{(k^*)}$ for each $(\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(k^*-1)})$ in a neighborhood of $(\widehat{\boldsymbol{\theta}}^{(0)}, \ldots, \widehat{\boldsymbol{\theta}}^{(k^*-1)})$. □

It follows from this assumption that the maximum value, $\max_{\boldsymbol{\theta}^{(k^*)} \in \Theta} \overline{\lambda}[V(\boldsymbol{x}^{(k^*)}, \boldsymbol{\theta}^{(k^*)})]$, is continuously differentiable in a neighborhood of $(\widehat{\boldsymbol{\theta}}^{(0)}, \ldots, \widehat{\boldsymbol{\theta}}^{(k^*-1)})$ and that the unique maximizing $\boldsymbol{\theta}^{(k^*)}$ is continuous there. The final assumption is now made.

Assumption 10. There exists at least one $k = 0, \ldots, k^* - 1$ such that the gradient of $\max_{\boldsymbol{\theta}^{(k^*)} \in \Theta} \overline{\lambda}[V(\boldsymbol{x}^{(k^*)}, \boldsymbol{\theta}^{(k^*)})]$ with respect to this $\boldsymbol{\theta}^{(k)}$ is nonzero at $(\widehat{\boldsymbol{\theta}}^{(0)}, \ldots, \widehat{\boldsymbol{\theta}}^{(k^*-1)})$. □

4. Stopping rule

This is the main section of this paper. In particular, we introduce a stopping rule into Algorithm 4 and show that the resulting algorithm always stops in a polynomial number of iterations and at its termination it either gives a probabilistic solution with high confidence or detects that there exists no deterministic solution in an approximated sense.

Before executing the algorithm to be proposed, we need to choose an initial candidate of a solution $\boldsymbol{x}^{(0)}$ and a step-size parameter $r > 0$ just as for Algorithm 4 in the previous section. In addition, we need to choose three numbers $R > r$, $0 < \epsilon < 1$, and $0 < \delta < 1$, where R is usually chosen large and the other two numbers are small. We use two counters in the algorithm: k counts the number of iterations and ℓ the number of updates. We define an integer

$$\overline{\ell} := \left\lceil \frac{R^2}{r^2} \right\rceil - 1$$

and a function

$$\kappa(\ell) := \left\lceil \left(\ln \frac{\pi^2 (\ell+1)^2}{6\delta} \right) \Big/ \ln \frac{1}{1-\epsilon} \right\rceil. \tag{1}$$

Algorithm 11.

0. Set $k := 0$ and $\ell := 0$.

1. If $\lambda^{(k-1)}, \lambda^{(k-2)}, \ldots, \lambda^{(k-\kappa(\ell))}$ are well-defined and are all negative, stop and give $\boldsymbol{x}^{(k)}$ as an output.

2. If ℓ reaches $\overline{\ell}$, stop with no output.

3. Randomly sample $\boldsymbol{\theta}^{(k)} \in \Theta$ according to the given probability measure P.

4. Check $\lambda^{(k)} := \overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)})]$ for negativity. Following the same rule as in Step 2 of Algorithm 4, update $\boldsymbol{x}^{(k)}$ if $\lambda^{(k)} \geq 0$ and keep $\boldsymbol{x}^{(k)}$ if $\lambda^{(k)} < 0$.

5. If $\lambda^{(k)} \geq 0$, set $\ell := \ell + 1$.

6. Set $k := k + 1$. Go back to Step 1. □

The next theorem, which is the main result of this paper, guarantees the performance of this algorithm. Namely, the statement (a) evaluates the computational complexity and the statements (b) and (c) give properties of the output. We denote by O_R an open ball with the center $\boldsymbol{x}^{(0)}$ and the radius R. Note that we do not need Assumptions 2, 3, or 7–10 any more. See Appendix B for the proof.

Theorem 12. *The following statements hold on Algorithm 11.*

(a) The number of iterations k is bounded as

$$k \leq \overline{\ell}\kappa(\overline{\ell}-1) = \overline{\ell}\left\lceil \left(\ln \frac{\pi^2\overline{\ell}^2}{6\delta}\right) \Big/ \ln \frac{1}{1-\epsilon} \right\rceil =: \overline{k}.$$

(b) If the algorithm stops at Step 2, the set $O_R \cap S$ does not contain an open ball of radius r.

(c) The probability that the algorithm stops at Step 1 but still the corresponding output $\boldsymbol{x}^{(k)}$ fails to satisfy $P\{\boldsymbol{\theta} \in \Theta : V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}) \prec O\} > 1-\epsilon$ is less than or equal to δ, where the probability is measured with respect to P^∞.

The statement (a) of Theorem 12 means that our algorithm stops in $\overline{k}$ iterations. The number $\overline{k}$ is of order $\mathrm{O}((\overline{\ell}/\epsilon)\ln(\overline{\ell}^2/\delta))$, which is a polynomial in R/r, $1/\epsilon$, and $\ln(1/\delta)$. Note that this number does not depend on the dimension of the parameter p. This forms a sharp contrast with deterministic algorithms whose complexity is usually of exponential order in p.

When the algorithm stops at Step 2, we see by the statement (b) that the set $O_R \cap S$ does not contain an open ball of radius r. This means that our choice of O_R, or equivalently, the choice of an initial solution candidate $\boldsymbol{x}^{(0)}$ and a radius R, is not appropriate or that the solution set S itself is too small. On the other hand, termination at Step 1 implies that the associated output is a probabilistic solution with high confidence. Note also that the algorithm always stops at Step 1 if $O_R \cap S$ includes an open ball of radius r.

Remark 13. Let us consider to replace $\kappa(\ell)$ in Algorithm 11 by a constant

$$\kappa' := \left\lceil \left(\ln \frac{\overline{\ell}}{\delta}\right) \Big/ \ln \frac{1}{1-\epsilon} \right\rceil.$$

Still we can derive the statements (b) and (c) of Theorem 12 and can prove (a) with a bound $k \leq \overline{\ell}\kappa'$, which is actually smaller than our bound $\overline{k}$. The reason why we did not employ κ' is that it depends on $\overline{\ell}$ and thus on R. Since we have only little knowledge on the location of the solution set in many cases, we need trial and error for an appropriate choice of the value of R. If we use κ' in the algorithm, we have to execute the algorithm from scratch every time we change the value of R. If we use $\kappa(\ell)$ instead, we do not have to do this because $\kappa(\ell)$ does not depend on R. □

Remark 14. Calafiore *et al.* [5] recently proposed, independently of the present author, the use of a stopping rule in a gradient-based and an ellipsoid-based randomized algorithms. One difference between their result and ours is that their stopping rule is based on the Chernoff bound and their bound on the number of iterations is of order $\mathrm{O}(1/\epsilon^2)$ while our bound is of order $\mathrm{O}(1/\epsilon)$. This difference is important because ϵ is chosen close to zero. Another difference is that they used κ' in the previous remark, which has a drawback stated there. □

Remark 15. The algorithms of [20, 6] used the subgradient of $\|V(\boldsymbol{x},\boldsymbol{\theta})^+\|_{\mathrm{F}}$ in place of $\overline{\lambda}[V(\boldsymbol{x},\boldsymbol{\theta})]$ for updating a solution candidate. Here the symbol $\|\cdot\|_{\mathrm{F}}$ stands for the Frobenius norm and the matrix $V(\boldsymbol{x},\boldsymbol{\theta})^+$ is the projection of $V(\boldsymbol{x},\boldsymbol{\theta})$ onto the cone of positive semidefinite matrices. The function $\|V(\boldsymbol{x},\boldsymbol{\theta})^+\|_{\mathrm{F}}$ resembles $\overline{\lambda}[V(\boldsymbol{x},\boldsymbol{\theta})]$ in that it is a convex function in $\boldsymbol{x}$ and it takes a nonpositive value if and only if $V(\boldsymbol{x},\boldsymbol{\theta}) \preceq O$. By this fact, it is possible to use $\|V(\boldsymbol{x},\boldsymbol{\theta})^+\|_{\mathrm{F}}$ in Algorithm 4 instead and to prove results corresponding to Proposition 5 and Theorem 6. A major difference is that one cannot distinguish $V(\boldsymbol{x},\boldsymbol{\theta}) \prec O$ from $V(\boldsymbol{x},\boldsymbol{\theta}) \preceq O$ by using $\|V(\boldsymbol{x},\boldsymbol{\theta})^+\|_{\mathrm{F}}$. Consequently, we cannot guarantee $V(\boldsymbol{x}^{(k)},\boldsymbol{\theta}) \prec O$ in Theorem 12 (c) with this function. This point is important because a strict matrix inequality is often required in control applications. □

5. Ellipsoid-based algorithm

The ellipsoid-based randomized algorithm of Kanev *et al.* [10] has properties and difficulties similar to those of the gradient-based one. We see in this section that introduction of a stopping rule and consideration of a probabilistic solution are again effective for resolution of the difficulties.

We first present the ellipsoid-based randomized algorithm in its original form. This algorithm iteratively updates an ellipsoid, which is expected to contain a deterministic solution. The ellipsoid at the kth iteration, $E^{(k)}$, is specified by its center $\boldsymbol{x}^{(k)}$ and a positive definite matrix $Q^{(k)}$ in the form of $\{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{x}-\boldsymbol{x}^{(k)})^{\mathrm{T}}(Q^{(k)})^{-1}(\boldsymbol{x}-\boldsymbol{x}^{(k)}) < 1\}$. Before executing the algorithm, we choose an initial ellipsoid $E^{(0)} = (\boldsymbol{x}^{(0)}, Q^{(0)})$. We need Assumption 2 as well as the following one.

Assumption 16. The intersection between the initial ellipsoid $E^{(0)}$ and the solution set S has a positive volume. □

Now the algorithm is presented.

Algorithm 17.

0. Set $k := 0$.

1. Randomly sample $\boldsymbol{\theta}^{(k)} \in \Theta$ according to the given probability measure P.

2. Check $\lambda^{(k)} := \overline{\lambda}[V(\boldsymbol{x}^{(k)},\boldsymbol{\theta}^{(k)})]$ for negativity.

   Case 1 The value $\lambda^{(k)}$ is nonnegative.
   Compute a subgradient, say $\boldsymbol{d}^{(k)}$, of $\overline{\lambda}[V(\boldsymbol{x}^{(k)},\boldsymbol{\theta}^{(k)})]$ as a convex function in $\boldsymbol{x}$. Update the current ellipsoid by setting

$$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} - \frac{Q^{(k)}\boldsymbol{d}^{(k)}}{(n+1)\sqrt{(\boldsymbol{d}^{(k)})^{\mathrm{T}}Q^{(k)}\boldsymbol{d}^{(k)}}},$$

$$Q^{(k+1)} := \frac{n^2}{n^2-1}\left[Q^{(k)} - \frac{2Q^{(k)}\boldsymbol{d}^{(k)}(\boldsymbol{d}^{(k)})^{\mathrm{T}}Q^{(k)}}{(n+1)(\boldsymbol{d}^{(k)})^{\mathrm{T}}Q^{(k)}\boldsymbol{d}^{(k)}}\right].$$

Case 2 The value $\lambda^{(k)}$ is negative.

Keep the current ellipsoid by setting $\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)}$ and $Q^{(k+1)} := Q^{(k)}$.

3. Set $k := k+1$. Go back to Step 1. □

On the performance of this algorithm, the next result is shown in [10].

Proposition 18. *Under Assumptions 2 and 16 there exists with probability one a finite k such that $\boldsymbol{x}^{(k)}$ produced by Algorithm 17 is a non-strict deterministic solution, where the probability is measured with P^∞.*

For each $\{\boldsymbol{\theta}^{(k)}\}$, we let k_{N} denote the minimum k with which $\boldsymbol{x}^{(k)}$ is a non-strict deterministic solution.

The above result, however, has difficulties similar to those of the original gradient-based algorithm. Indeed, it is difficult to choose an initial ellipsoid $E^{(0)}$ so that it satisfies Assumption 16; It is difficult to know when $\boldsymbol{x}^{(k)}$ arrives at a non-strict deterministic solution; The number of necessary iterations k_{N} has infinite expectation as in the following theorem. The proof of the theorem is omitted because it is essentially the same as that of Theorem 6.

Theorem 19. *Suppose that an initial ellipsoid $E^{(0)}$ is provided. Under Assumptions 7–10, the number of necessary iterations k_{N} in Algorithm 17 has infinite expectation, where the expectation is taken with the measure P^∞.*

In order to resolve these difficulties we consider the following modified algorithm with a stopping rule. Before executing the algorithm, we choose not only an initial ellipsoid $E^{(0)}$ but also three numbers $0 < \mu$, $0 < \epsilon < 1$, and $0 < \delta < 1$. These numbers are usually chosen close enough to zero. We define

$$\overline{\ell} := \left\lceil 2(n+1)\,\ln\frac{\mathrm{vol}\,E^{(0)}}{\mu}\right\rceil \tag{2}$$

and $\kappa(\ell)$ as in (1).

Algorithm 20.

0. Set $k := 0$ and $\ell := 0$.

1. If $\lambda^{(k-1)}, \lambda^{(k-2)}, \ldots, \lambda^{(k-\kappa(\ell))}$ are well-defined and are all negative, stop and give $\boldsymbol{x}^{(k)}$ as an output.

2. If ℓ reaches $\overline{\ell}$, stop with no output.

3. Randomly sample $\boldsymbol{\theta}^{(k)} \in \Theta$ according to the given probability measure P.

4. Check $\lambda^{(k)} := \overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)})]$ for negativity. Following the same rule as in Step 2 of Algorithm 17, update $E^{(k)}$ if $\lambda^{(k)} \geq 0$ and keep $E^{(k)}$ if $\lambda^{(k)} < 0$.

5. If $\lambda^{(k)} \geq 0$, set $\ell := \ell + 1$.

6. Set $k := k + 1$. Go back to Step 1. □

The next theorem evaluates the computational complexity of this algorithm and guarantees the quality of its output. Note that Assumption 2 or 16 is not necessary any more. The proof is found in Appendix C.

Theorem 21. *The following statements hold on Algorithm 20.*

(a) The number of iterations k is bounded as

$$k \leq \overline{\ell}\kappa(\overline{\ell} - 1) = \overline{\ell}\left\lceil \left(\ln \frac{\pi^2 \overline{\ell}^2}{6\delta} \right) \Big/ \ln \frac{1}{1-\epsilon} \right\rceil =: \overline{k}.$$

(b) If the algorithm stops at Step 2, the volume of the set $E^{(0)} \cap S$ is less than or equal to μ.

(c) The probability that the algorithm stops at Step 1 but still the corresponding output $\boldsymbol{x}^{(k)}$ fails to satisfy $P\{\boldsymbol{\theta} \in \Theta : V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}) \prec O\} > 1 - \epsilon$ is less than or equal to δ, where the probability is measured with respect to P^∞.

The statement (a) of this theorem gives an upper bound of the number of iterations, which is of polynomial order in n, $\ln(\mathrm{vol}\, E^{(0)}/\mu)$, $1/\epsilon$, and $\ln(1/\delta)$. When the algorithm stops at Step 2, we have $\mathrm{vol}\,(E^{(0)} \cap S) \leq \mu$ by the statement (b). This means that our choice of the initial ellipsoid $E^{(0)}$ is not good or the solution set S itself is too small. On the other hand, termination at Step 1 implies that the associated output is a probabilistic solution with high confidence.

Remarks 13–15 also apply *mutatis mutandis*.

6. Optimization

Algorithms 11 and 20 in the preceding sections can be adapted to optimization problems by their bisectional use. This section presents this result on the ellipsoid-based randomized algorithm. The corresponding result on the gradient-based algorithm is easily obtained and hence omitted.

We consider a symmetric-matrix-valued function $V_\gamma(\boldsymbol{x}, \boldsymbol{\theta})$ depending on a performance index $\gamma \in \mathbb{R}$, where $\boldsymbol{x} \in \mathbb{R}^n$ is a variable and $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$ is a parameter. Although we allow nonlinear

dependence of V_γ on γ, we assume its monotonicity in the sense that $\gamma_1 < \gamma_2$ and $V_{\gamma_1}(\boldsymbol{x}, \boldsymbol{\theta}) \prec O$ imply $V_{\gamma_2}(\boldsymbol{x}, \boldsymbol{\theta}) \prec O$. The problem to be considered is the following.

Problem 22. Obtain the infimum of $\gamma \in \mathbb{R}$ under the condition that there exists $\boldsymbol{x} \in \mathbb{R}^n$ satisfying $V_\gamma(\boldsymbol{x}, \boldsymbol{\theta}) \prec O$ for all $\boldsymbol{\theta} \in \Theta$. □

This problem naturally arises if we consider optimization for some performance index, say H^2 or H^∞, in design of robust and gain-scheduled control systems.

We consider to solve this problem probabilistically. The basic idea is the following: we look for a probabilistic solution with some γ using Algorithm 20; when we succeed, we try again with a smaller γ; when we fail, we try with a larger γ. Before executing the algorithm, we choose three numbers $\underline{\gamma} < \overline{\gamma}$ and $\gamma_{\mathrm{tol}} > 0$, where $(\underline{\gamma}, \overline{\gamma})$ is the interval in which we look for an optimal γ and the number γ_{tol} stands for a tolerable error level. We define a number $\overline{m}$ and a function $\widetilde{\kappa}(\ell)$ by

$$\overline{m} := \lceil \log_2[(\overline{\gamma} - \underline{\gamma})/\gamma_{\mathrm{tol}}] \rceil,$$
$$\widetilde{\kappa}(\ell) := \left\lceil \left(\ln \frac{\pi^2 \overline{m} (\ell+1)^2}{6\delta} \right) \Big/ \ln \frac{1}{1-\epsilon} \right\rceil.$$

As in the previous section, we choose in advance an initial ellipsoid $E^{(0)}$ and three numbers μ, ϵ, and δ, which are to be used in Algorithm 20 that works as a subroutine in the following algorithm.

Algorithm 23.

0. Set $\gamma_{\mathrm{U}} := \overline{\gamma}$ and $\gamma_{\mathrm{L}} := \underline{\gamma}$.
1. If $\gamma_{\mathrm{U}} - \gamma_{\mathrm{L}} \leq \gamma_{\mathrm{tol}}$, stop the algorithm.
2. Set $\gamma := (\gamma_{\mathrm{U}} - \gamma_{\mathrm{L}})/2$. Execute Algorithm 20 with replacing $V(\boldsymbol{x}, \boldsymbol{\theta})$ by $V_\gamma(\boldsymbol{x}, \boldsymbol{\theta})$ and $\kappa(\ell)$ by $\widetilde{\kappa}(\ell)$.
3. If Algorithm 20 stops at Step 1, set $\gamma_{\mathrm{U}} := \gamma$. If it stops at Step 2, set $\gamma_{\mathrm{L}} := \gamma$.
4. Go back to Step 1. □

The performance of this algorithm is stated in the next theorem, whose proof is found in Appendix D. In particular, the statement (a) gives the computational complexity of the algorithm; the statement (b) says that, with γ_{L} at the termination of the algorithm, there is no deterministic solution in an approximated sense; the statement (c) says that, with γ_{U} at the termination of the algorithm, a probabilistic solution is found. Since $\gamma_{\mathrm{U}} - \gamma_{\mathrm{L}} \leq \gamma_{\mathrm{tol}}$, our optimization problem 22 is approximately solved.

Theorem 24. *The following statements hold on Algorithm 23.*

(a) *Algorithm 20 is executed* $\overline{m}$ *times and the total sum of its iteration numbers is less than or equal to*

$$\overline{m}\overline{\ell}\widetilde{\kappa}(\overline{\ell}-1)=\overline{m}\overline{\ell}\left\lceil\left(\ln\frac{\pi^2\overline{m}\overline{\ell}^2}{6\delta}\right)\Big/\ln\frac{1}{1-\epsilon}\right\rceil$$

with $\overline{\ell}$ *in (2).*

(b) *If* $\gamma_{\mathrm{L}}\neq\underline{\gamma}$ *at the termination of Algorithm 23, the volume of the set* $E^{(0)}\cap S_{\gamma_{\mathrm{L}}}$ *is less than or equal to* μ*, where* $S_\gamma:=\{\boldsymbol{x}\in\mathbb{R}^n : V_\gamma(\boldsymbol{x},\boldsymbol{\theta})\prec O \text{ for all } \boldsymbol{\theta}\in\Theta\}$.

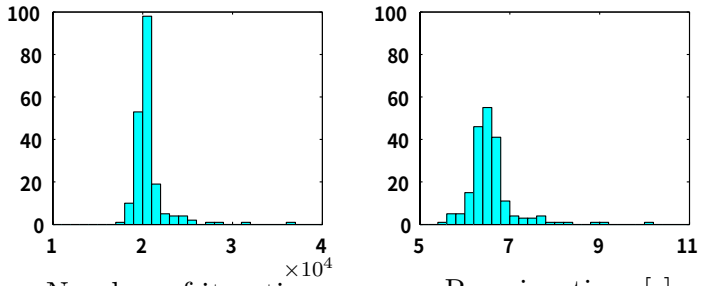
(c) *The probability that* $\gamma_{\mathrm{U}}\neq\overline{\gamma}$ *holds at the termination of Algorithm 23 and the output of Algorithm 20 corresponding to* $\gamma=\gamma_{\mathrm{U}}$ *is not a probabilistic solution is less than or equal to* δ*, where the probability is measured according to* P^∞ *for all the parameter values* $\{\boldsymbol{\theta}^{(k)}\}$ *sampled in the executions of Algorithm 20.*

The bound on the total number of iterations given in the statement (a) is of polynomial order in n, $\ln(\mathrm{vol}\,E^{(0)}/\mu)$, $1/\epsilon$, $\ln(1/\delta)$, and $\log_2[(\overline{\gamma}-\underline{\gamma})/\gamma_{\mathrm{tol}}]$. This is an attractive property, even though optimization is done only in a weak sense, because we are solving a difficult optimization problem with infinitely many constraints. Calafiore and Campi [4] provided a different approach to Problem 22. Although their approach solves the problem in a stronger sense than ours, it is computationally more demanding. Indeed, their approach typically requires us to solve semidefinite programming problems with more than thousands of constraints.

7. Numerical example

In order to illustrate our approach, we consider robust stabilization of a tower crane model, which is taken from [23] with some simplification. Application of Algorithm 20 successfully gave a probabilistic solution as we will see below, which shows practical usefulness of our approach. Algorithm 11 is also applicable though its result is omitted.

The considered plant is described by a fourth-order state-space representation and its coefficient matrices nonlinearly depend on a three-dimensional parameter $\boldsymbol{\theta}$, which expresses the rope length, the boom angle at the equilibrium point, and the load weight. It is supposed that each component of $\boldsymbol{\theta}$ can take any value in some prescribed interval, which means that our parameter set Θ is a hyper rectangle. See [17] for more details of this plant. We consider design of a state feedback law that robustly stabilizes the plant. We formulate this problem in the form of Problem 1 by using a parameter-independent Lyapunov function and apply Algorithm 20. We choose the measure P to be the uniform distribution on Θ and set the initial ellipsoid $E^{(0)}$ by



(a) (b)

Figure 1. Histograms for 200 executions of Algorithm 20

letting $\boldsymbol{x}^{(0)}$ be the zero vector and the matrix $R^{(0)}$ be the identity. The parameters are set as $\mu = 10^{-9}\mathrm{vol}\, E^{(0)}$, $\epsilon = 0.001$, and $\delta = 0.0001$.

We tried 200 executions of Algorithm 20 with Pentium 4 of 2.4GHz and memory of 2.0GByte. In every execution, the algorithm stopped at Step 1 and gave an output. Figure 1 (a) shows the histogram of the number of iterations and (b) shows that of the running time. While Theorem 21 (a) guarantees that our algorithm stops in $\overline{k} = 1.40 \times 10^7$ iterations, Figure 1 (a) shows that the algorithm actually stopped in much smaller number of iterations. Moreover, Figure 1 (b) shows that the running time is less than seven seconds in many cases. From these results, we can see that the proposed algorithm is computationally acceptable. By Theorem 21 (c), on the other hand, the probability that the algorithm stops at Step 1 but the obtained output is not a probabilistic solution of $\epsilon = 0.001$ is less than or equal to $\delta = 0.0001$. This means that the obtained output is a probabilistic solution with high confidence.

Recall that the ellipsoid randomized algorithm in the original form does not have a stopping rule. It is hence difficult to know when to stop the iteration. If we stop it with an *ad hoc* rule, meaning of the resulting output is not as clear as above.

8. Conclusion

In this paper, difficulties of the gradient-based and the ellipsoid-based randomized algorithms are discussed and a technique for their resolution is presented. Namely, by introducing stopping rules into those algorithms, it is possible to bound the number of necessary iterations by a polynomial in the problem size and to show that the algorithms either provide a probabilistic solution with high confidence or detect approximate infeasibility of the problem. It is also possible to adapt the algorithms for finding an optimal solution of a parameter-dependent LMI.

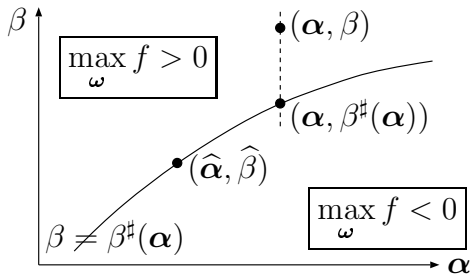


Figure 2. The function $\beta^\sharp(\boldsymbol{\alpha})$

Acknowledgments

The author is thankful to the following people for their comments: Stephen Boyd, Marco Campi, Fabrizio Dabbene, Shinji Hara, Satoru Iwata, Masakazu Kojima, Kazuo Murota, Maria Prandini, Roberto Tempo, and Takashi Tsuchiya. In particular, the contents of Section 6 are based on the advice of Masakazu Kojima. Part of this work is supported by the 21st Century COE Program on Information Science and Technology Strategic Core and a Grant-in-Aid of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

A. Proof of Theorem 6

We prepare some notation. As was noticed after Assumption 8, the eigenvalue $\overline{\lambda}[V(\boldsymbol{x}^{(k^*)}, \boldsymbol{\theta}^{(k^*)})]$ is regarded as a function of $(\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(k^*)})$, which is continuously differentiable in a neighborhood of $(\widehat{\boldsymbol{\theta}}^{(0)}, \ldots, \widehat{\boldsymbol{\theta}}^{(k^*)})$. Let us write this function as $f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega})$, where $[\boldsymbol{\alpha}^{\mathrm{T}} \quad \beta]^{\mathrm{T}}$ stands for $[(\boldsymbol{\theta}^{(0)})^{\mathrm{T}} \quad \ldots \quad (\boldsymbol{\theta}^{(k^*-1)})^{\mathrm{T}}]^{\mathrm{T}}$ with β being the last element of the vector and $\boldsymbol{\omega}$ stands for $\boldsymbol{\theta}^{(k^*)}$. We also use the notation $[\widehat{\boldsymbol{\alpha}}^{\mathrm{T}} \quad \widehat{\beta}]^{\mathrm{T}} = [(\widehat{\boldsymbol{\theta}}^{(0)})^{\mathrm{T}} \quad \ldots \quad (\widehat{\boldsymbol{\theta}}^{(k^*-1)})^{\mathrm{T}}]^{\mathrm{T}}$ and $\widehat{\boldsymbol{\omega}} = \widehat{\boldsymbol{\theta}}^{(k^*)}$. By rearranging the order of the elements if necessary, we can assume due to Assumption 10 that $(\partial/\partial\beta) \max_{\boldsymbol{\omega}\in\Theta} f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega}) \neq 0$ at $(\widehat{\boldsymbol{\alpha}}, \widehat{\beta})$. Using the implicit function theorem we can define a continuously differentiable function $\beta^\sharp(\boldsymbol{\alpha})$ so that $\beta^\sharp(\widehat{\boldsymbol{\alpha}})$ is equal to $\widehat{\beta}$ and $\max_{\boldsymbol{\omega}\in\Theta} f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega}) = 0$ is equivalent to $\beta = \beta^\sharp(\boldsymbol{\alpha})$ in a neighborhood of $(\widehat{\boldsymbol{\alpha}}, \widehat{\beta})$. The situation is illustrated in Figure 2.

For each pair of $(\boldsymbol{\alpha}, \beta)$, a point $\boldsymbol{x}^{(k^*)}$ is determined. A pair $(\boldsymbol{\alpha}, \beta)$ satisfies $\max_{\boldsymbol{\omega}\in\Theta} f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega}) \geq 0$ if and only if the corresponding $\boldsymbol{x}^{(k^*)}$ does not belong to the solution set S, in which case this $\boldsymbol{x}^{(k^*)}$ needs to be further updated. We will evaluate the probability of this update, which is equal to $P^\infty\{\boldsymbol{\omega} \in \Theta : f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega}) \geq 0\}$. Indeed, the two lemmas below show that this probability is bounded by a linear function of $|\beta - \beta^\sharp(\boldsymbol{\alpha})|$. Once such evaluation is obtained, the theorem is easily proved.

We prepare one more function before presenting the first lemma. By Assumption 9, the maximum value, $\max_{\boldsymbol{\omega}\in\Theta} f(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega})$, is attained at a unique $\boldsymbol{\omega}$, which is written as $\boldsymbol{\omega}^\sharp(\boldsymbol{\alpha})$. This

$\boldsymbol{\omega}^\sharp(\boldsymbol{\alpha})$ is continuous in $\boldsymbol{\alpha}$ as was noted after Assumption 9. Moreover, $\nabla_{\boldsymbol{\omega}} f(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha}))$, which is the gradient of $f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega})$ with respect to $\boldsymbol{\omega}$ evaluated at $(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha}))$, is a continuous function of $\boldsymbol{\alpha}$ in a neighborhood of $\widehat{\boldsymbol{\alpha}}$ and is nonzero there by Assumption 8 (d).

Lemma 25. *There exist a neighborhood of $(\widehat{\boldsymbol{\alpha}}, \widehat{\beta})$ and a positive number c such that, if $(\boldsymbol{\alpha}, \beta)$ belongs to that neighborhood and $f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega}) = 0$ holds, there also holds either*

$$\left|[\nabla_{\boldsymbol{\omega}} f(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha}))]^{\mathrm{T}}[\boldsymbol{\omega} - \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha})]\right| < c|\beta - \beta^\sharp(\boldsymbol{\alpha})|$$

or $(\beta, \boldsymbol{\omega}) = (\beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha}))$.

Proof. Consider $(\partial/\partial\beta) f(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha}))$, which is a partial derivative of $f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega})$ with respect to β evaluated at $(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha}))$. This partial derivative is continuous as a function of $\boldsymbol{\alpha}$, which implies the existence of a positive number c that satisfies $\left|(\partial/\partial\beta) f(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha}))\right| < c$ in a neighborhood of $\widehat{\boldsymbol{\alpha}}$. We choose and fix one $\boldsymbol{\alpha}$ in this neighborhood and show that the lemma holds with the above c. To this end, we assume the contrary and derive a contradiction.

With fixing one $\boldsymbol{\alpha}$ in the neighborhood of $\widehat{\boldsymbol{\alpha}}$, we write $f(\beta, \boldsymbol{\omega})$, $\beta^\sharp$, and $\boldsymbol{\omega}^\sharp$ to mean $f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega})$, $\beta^\sharp(\boldsymbol{\alpha})$, and $\boldsymbol{\omega}^\sharp(\boldsymbol{\alpha})$, respectively. Suppose that there exist sequences $\{\beta^j\}$ and $\{\boldsymbol{\omega}^j\}$ that satisfy $\beta^j \to \beta^\sharp$ $(j \to \infty)$, $\beta^j \neq \beta^\sharp$, $f(\beta^j, \boldsymbol{\omega}^j) = 0$, and $\left|[\nabla_{\boldsymbol{\omega}} f(\beta^\sharp, \boldsymbol{\omega}^\sharp)]^{\mathrm{T}}[\boldsymbol{\omega}^j - \boldsymbol{\omega}^\sharp]\right| \geq c|\beta^j - \beta^\sharp|$. Since the parameter set Θ is compact, a subsequence of $\{\boldsymbol{\omega}^j\}$ converges to a point in Θ, which has to be $\boldsymbol{\omega}^\sharp$ by its assumed uniqueness. Hence we can assume that $\{\boldsymbol{\omega}^j\}$ converges to $\boldsymbol{\omega}^\sharp$. Expand $f(\beta, \boldsymbol{\omega})$ around $(\beta^\sharp, \boldsymbol{\omega}^\sharp)$ to obtain

$$0 = f(\beta^j, \boldsymbol{\omega}^j) = \frac{\partial f(\beta^\sharp, \boldsymbol{\omega}^\sharp)}{\partial \beta}(\beta^j - \beta^\sharp) + \left[\nabla_{\boldsymbol{\omega}} f(\beta^\sharp, \boldsymbol{\omega}^\sharp)\right]^{\mathrm{T}}(\boldsymbol{\omega}^j - \boldsymbol{\omega}^\sharp) + g(\beta^j, \boldsymbol{\omega}^j),$$

where $g(\beta^j, \boldsymbol{\omega}^j)$ is a function that converges to zero faster than $|\beta^j - \beta^\sharp|$ and $\|\boldsymbol{\omega}^j - \boldsymbol{\omega}^\sharp\|$ as $j \to \infty$. This equation further gives

$$\left|\frac{\partial f(\beta^\sharp, \boldsymbol{\omega}^\sharp)}{\partial \beta}\right| = \left|\frac{\left[\nabla_{\boldsymbol{\omega}} f(\beta^\sharp, \boldsymbol{\omega}^\sharp)\right]^{\mathrm{T}}(\boldsymbol{\omega}^j - \boldsymbol{\omega}^\sharp)}{\beta^j - \beta^\sharp} + \frac{g(\beta^j, \boldsymbol{\omega}^j)}{\beta^j - \beta^\sharp}\right|,$$

whose right-hand side is larger than or equal to c in the limit of $j \to \infty$. This contradicts the definition of c. □

Lemma 26. *There exist a neighborhood of $(\widehat{\boldsymbol{\alpha}}, \widehat{\beta})$ and a positive number c' such that all $(\boldsymbol{\alpha}, \beta)$ in that neighborhood satisfy $P\{\boldsymbol{\omega} \in \Theta : f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega}) \geq 0\} \leq c'|\beta - \beta^\sharp(\boldsymbol{\alpha})|$.*

Proof. In Lemma 25, a neighborhood of $(\widehat{\boldsymbol{\alpha}}, \widehat{\beta})$ can be chosen to be convex without loss of generality. Suppose that $f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega}) \geq 0$ holds for some $(\boldsymbol{\alpha}, \beta)$ in that neighborhood and $\boldsymbol{\omega} \in \Theta$. Since $f(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}) \leq 0$, continuity of $f(\boldsymbol{\alpha}, \beta, \boldsymbol{\omega})$ implies that there exists $\widetilde{\beta}$ satisfying

$f(\boldsymbol{\alpha}, \widetilde{\beta}, \boldsymbol{\omega}) = 0$ on the line segment connecting β and $\beta^\sharp(\boldsymbol{\alpha})$. From Lemma 25 it follows that $\left|[\nabla_{\boldsymbol{\omega}} f(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha}))]^{\mathrm{T}}[\boldsymbol{\omega} - \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha})]\right| \leq c|\widetilde{\beta} - \beta^\sharp(\boldsymbol{\alpha})| \leq c|\beta - \beta^\sharp(\boldsymbol{\alpha})|$. Due to Assumption 7, the probability of $\boldsymbol{\omega} \in \Theta$ satisfying this inequality is bounded by $c'|\beta - \beta^\sharp(\boldsymbol{\alpha})|$ for some $c' > 0$. Note that we can choose the same c' for all $\boldsymbol{\alpha}$ in a neighborhood of $\widehat{\boldsymbol{\alpha}}$, because the gradient $\nabla_{\boldsymbol{\omega}} f(\boldsymbol{\alpha}, \beta^\sharp(\boldsymbol{\alpha}), \boldsymbol{\omega}^\sharp(\boldsymbol{\alpha}))$ is continuous and nonzero there. □

We now prove the theorem. We can assume without loss of generality that $(\partial/\partial\beta) \max_{\boldsymbol{\omega}\in\Theta} f(\widehat{\boldsymbol{\alpha}}, \widehat{\beta}, \boldsymbol{\omega})$ is not only nonzero but also positive. We choose positive numbers a and b small enough that the set $A = \{(\boldsymbol{\alpha}, \beta) : \|\boldsymbol{\alpha} - \widehat{\boldsymbol{\alpha}}\| \leq a \text{ and } \beta^\sharp(\boldsymbol{\alpha}) \leq \beta \leq \beta^\sharp(\boldsymbol{\alpha}) + b\}$ is contained in the neighborhood of Lemma 26. For a pair $(\boldsymbol{\alpha}, \beta) \in A$, the corresponding $\boldsymbol{x}^{(k^*)}$ does not belong to the solution set S and thus needs to be updated. Because the probability of update is bounded by $c'|\beta - \beta^\sharp(\boldsymbol{\alpha})|$ due to Lemma 26, the expected number of iterations required to make one update is larger than or equal to $1/c'|\beta - \beta^\sharp(\boldsymbol{\alpha})|$. If we integrate this number in A with P^∞, it gives a lower bound of the expected number of iterations necessary to find a non-strict deterministic solution. The result is infinity, which completes the proof.

B. Proof of Theorem 12

(a) By the construction of the algorithm, at most $\kappa(\ell_0)$ iterations are made during the period that the counter ℓ has the value ℓ_0. Since the algorithm stops when ℓ reaches $\overline{\ell}$, the number of iterations k has to be less than or equal to $\kappa(0) + \cdots + \kappa(\overline{\ell} - 1) \leq \overline{\ell}\kappa(\overline{\ell} - 1)$.

(b) Suppose that the set $O_R \cap S$ includes an open ball of radius r. If we can show that the algorithm cannot stop at Step 2 in this case, the proof is complete. Let $\boldsymbol{x}^*$ denote the center of the above ball of radius r. By definition, we have $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|^2 \leq (R - r)^2 < R^2 - r^2$. If $\boldsymbol{x}^{(k+1)}$ is obtained by an update of $\boldsymbol{x}^{(k)}$, there holds $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|^2 \leq \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|^2 - r^2$ owing to Theorem 1 of [20] and Theorem 1 of [6]. Now suppose that $\boldsymbol{x}^{(k)}$ has experienced $\overline{\ell} - 1$ updates since $\boldsymbol{x}^{(0)}$. Then there holds $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|^2 \leq \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|^2 - (\overline{\ell} - 1)r^2 < R^2 - \overline{\ell}r^2$. Because the right-hand side is less than or equal to r^2 by the definition of $\overline{\ell}$, the candidate $\boldsymbol{x}^{(k)}$ belongs to the solution set S. This implies that $\boldsymbol{x}^{(k)}$ can never be updated and thus ℓ never reaches $\overline{\ell}$. The proof is complete.

(c) This is an extension of a result of [11] and [24]. Write the solution candidate after ℓ updates as $\boldsymbol{x}^{[\ell]}$ noting that the candidate remains unchanged until the next update. We define two events for each of $\ell = 0, 1, \ldots$:

M_ℓ : The number of updates reaches ℓ and the candidate $\boldsymbol{x}^{[\ell]}$ is not updated for consecutive $\kappa(\ell)$ iterations;

B_ℓ : The number of updates reaches ℓ and $\boldsymbol{x}^{[\ell]}$ satisfies $P\{\boldsymbol{\theta} \in \Theta : V(\boldsymbol{x}^{[\ell]}, \boldsymbol{\theta}) \prec O\} \leq 1 - \epsilon$.

In the following, we will show

$$P^{\infty}\big[(M_0 \cap B_0) \cup (M_1 \cap B_1) \cup \cdots\big] \le \delta, \tag{3}$$

which establishes the claim.

We have, for any ℓ,

$$P^{\infty}(M_\ell \cap B_\ell) = P^{\infty}(M_\ell|B_\ell)P^{\infty}(B_\ell) \le P^{\infty}(M_\ell|B_\ell) \le (1-\epsilon)^{\kappa(\ell)},$$

where $P^{\infty}(M_\ell|B_\ell)$ expresses the conditional probability of M_ℓ with B_ℓ being assumed. Therefore, we have

$$\begin{aligned} P^{\infty}\big[(M_0 \cap B_0) \cup (M_1 \cap B_1) \cup \cdots\big] &\le (1-\epsilon)^{\kappa(0)} + (1-\epsilon)^{\kappa(1)} + \cdots \\ &\le \frac{6\delta}{\pi^2}\Big(1 + \frac{1}{2^2} + \cdots\Big) = \delta, \end{aligned}$$

which shows (3).

C. Proof of Theorem 21

Because we can prove the statements (a) and (c) similarly to the corresponding statements of Theorem 12, we present only the proof of (b).

Suppose that the ellipsoid $E^{(k)}$ is updated. Then, the ellipsoid $E^{(k+1)}$ is in fact the minimum-volume ellipsoid that contains the intersection between the original ellipsoid $E^{(k)}$ and the half space $\{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{d}^{(k)})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^{(k)}) < 0\}$ [3]. As we will see below, this half space includes the solution set S. Therefore $E^{(k+1)}$ includes the set $E^{(k)} \cap S$. It can also be shown that $\operatorname{vol} E^{(k+1)} < (\operatorname{vol} E^{(k)})\mathrm{e}^{-1/2(n+1)}$ [3]. Hence, if the ellipsoid $E^{(k)}$ has experienced $\overline{\ell}$ updates, it satisfies $E^{(k)} \supseteq E^{(0)} \cap S$ and $\operatorname{vol} E^{(k)} < (\operatorname{vol} E^{(0)})\mathrm{e}^{-\overline{\ell}/2(n+1)} \le \mu$, which show the claim.

We show that the half space $\{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{d}^{(k)})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^{(k)}) < 0\}$ includes the solution set S. To this end, note that $\overline{\lambda}[V(\boldsymbol{x}, \boldsymbol{\theta}^{(k)})] \ge \overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)})] + (\boldsymbol{d}^{(k)})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^{(k)}) \ge (\boldsymbol{d}^{(k)})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^{(k)})$ because $\boldsymbol{d}^{(k)}$ is the subgradient of $\overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)})]$ and the value $\overline{\lambda}[V(\boldsymbol{x}^{(k)}, \boldsymbol{\theta}^{(k)})]$ is nonnegative. By this inequality, if $\boldsymbol{x}$ belongs to S, it satisfies $\overline{\lambda}[V(\boldsymbol{x}, \boldsymbol{\theta}^{(k)})] < 0$ and thus $(\boldsymbol{d}^{(k)})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^{(k)}) < 0$.

D. Proof of Theorem 24

(a) By construction of the algorithm, the number of executions of Algorithm 20 is the minimum m that satisfies $2^{-m}(\overline{\gamma} - \underline{\gamma}) \le \gamma_{\mathrm{tol}}$, which is equal to $\overline{m}$. The number of iterations in each execution of Algorithm 20 is bounded by $\overline{\ell}\widetilde{\kappa}(\overline{\ell} - 1)$ by the same reasoning as in the proof of Theorem 12 (a).

(b) The relation $\gamma_{\mathrm{L}} \ne \underline{\gamma}$ means that, when Algorithm 20 ran with $\gamma = \gamma_{\mathrm{L}}$, it stopped at Step 2 with giving no output. Thus there holds $\operatorname{vol}(E^{(0)} \cap S_{\gamma_{\mathrm{L}}}) \le \mu$.

(c) Consider the mth execution of Algorithm 20 and define the events M_ℓ^m and B_ℓ^m as in the proof of Theorem 12 (c). What we want to show is that

$$P^\infty\Big[\bigcup_{m=1}^{\overline{m}}\bigcup_{\ell=0,1,\ldots}(M_\ell^m \cap B_\ell^m)\Big] \leq \delta.$$

This left-hand side is bounded by the sum $\sum_{m=1}^{\overline{m}} P^\infty\big[\bigcup_{\ell=0,1,\ldots}(M_\ell^m \cap B_\ell^m)\big]$ and each of its terms is bounded by $\delta/\overline{m}$ by the reasoning in the proof of Theorem 12 (c). Now the proof is complete.

References

[1] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H^∞ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.

[2] G. Becker and A. Packard, "Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback," *Systems & Control Letters*, vol. 23, no. 3, pp. 205–215, 1994.

[3] R. G. Bland, D. Goldfarb, and M. J. Todd, "The ellipsoid method: A survey," *Operations Research*, vol. 29, no. 6, pp. 1039–1091, 1981.

[4] G. Calafiore and M. C. Campi, "Uncertain convex programs: Randomized solutions and confidence levels," *Mathematical Programming*, to appear.

[5] G. Calafiore, F. Dabbene, and R. Tempo, "Randomized algorithms in robust control," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, USA, December 2003, pp. 1908–1913.

[6] G. Calafiore and B. Polyak, "Stochastic algorithms for exact and approximate feasibility of robust LMIs," *IEEE Transactions on Automatic Control*, vol. 46, no. 11, pp. 1755–1759, 2001.

[7] Y. Fujisaki, F. Dabbene, and R. Tempo, "Probabilistic design of LPV control systems," *Automatica*, vol. 39, no. 8, pp. 1323–1337, 2003.

[8] P. Gahinet, P. Apkarian, and M. Chilali, "Affine parameter-dependent Lyapunov functions and real parametric uncertainty," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 436–442, 1996.

[9] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*. Berlin, Germany: Springer, 2001.

[10] S. Kanev, B. De Schutter, and M. Verhaegen, "An ellipsoid algorithm for probabilistic robust controller design," *Systems & Control Letters*, vol. 49, no. 5, pp. 365–375, 2003.

[11] P. Khargonekar and A. Tikku, "Randomized algorithms for robust control analysis and synthesis have polynomial complexity," in *Proceedings of the 35th IEEE Conference on Decision and Control*, Kobe, Japan, December 1996, pp. 3470–3475.

[12] D. Liberzon and R. Tempo, "Gradient algorithms for finding common Lyapunov functions," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, USA, December 2003, pp. 4782–4787.

[13] A. Ohara and Y. Sasaki, "On solvability and numerical solutions of parameter-dependent differential matrix inequality," in *Proceedings of the First SICE Annual Conference*, Kyoto, Japan, May 2001, pp. 61–64.

[14] Y. Oishi, "An ellipsoid randomized algorithm for the use of non-common Lyapunov functions," in *Proceedings of the Third SICE Annual Conference on Control Systems*, Kobe, Japan, May 2003, pp. 473–476.

[15] Y. Oishi, "A probabilistic approach to robust control" (in Japanese), in *Proceedings of the 32nd SICE Symposium on Control Theory*, Hamamatsu, Japan, November 2003, pp. 13–18.

[16] Y. Oishi, "Probabilistic design of a robust state-feedback controller based on parameter-dependent Lyapunov functions," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, USA, December 2003, pp. 1920–1925.

[17] Y. Oishi and H. Kimura, "Randomized algorithms to solve parameter-dependent linear matrix inequalities," *Technical Report METR 2001-04*, Department of Mathematical Informatics, The University of Tokyo, Japan, 2001.

[18] Y. Oishi and H. Kimura, "Computational complexity of randomized algorithms for solving parameter-dependent linear matrix inequalities," *Automatica*, vol. 39, no. 12, pp. 2149–2156, 2003.

[19] B. T. Polyak, "Random algorithms for solving convex inequalities," in *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, D. Butnariu, Y. Censor, and S. Reich, Eds. Amsterdam, The Netherlands: Elsevier, 2001, pp. 409–422.

[20] B. T. Polyak and R. Tempo, "Probabilistic robust design with linear quadratic regulators," *Systems & Control Letters*, vol. 43, pp. 343–353, 2001.

[21] C. W. Scherer, "Higher-order relaxations for robust LMI problems with verifications for exactness," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, USA, December 2003, pp. 4652–4657.

[22] A. N. Shiryaev, *Probability*, second ed. New York, USA: Springer, 1996.

[23] K. Takagi and H. Nishimura, "Gain-scheduled control of a tower crane considering varying load-rope length," *JSME International Journal*, Ser. C, vol. 42, no. 4, pp. 914–921, 1999.

[24] R. Tempo, E. W. Bai, and F. Dabbene, "Probabilistic robustness analysis: explicit bounds for the minimum number of samples," *Systems & Control Letters*, vol. 30, no. 5, pp. 237–242, 1997.

[25] R. Watanabe, K. Uchida, M. Fujita, and E. Shimemura, "L^2 gain and H^∞ control of linear systems with scheduling parameter," in *Proceedings of the 33rd IEEE Conference on Decision and Control*, Lake Buena Vista, USA, December 1994, pp. 1412–1414.