

Technical report CSE02-008

The ellipsoid algorithm for probabilistic robust controller design*

S. Kanev, B. De Schutter, and M. Verhaegen

If you want to cite this report, please use the following reference instead:

S. Kanev, B. De Schutter, and M. Verhaegen, “The ellipsoid algorithm for probabilistic robust controller design,” *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada, pp. 2248–2253, Dec. 2002.

The Ellipsoid Algorithm for Probabilistic Robust Controller Design¹

S. Kanev[‡], B. De Schutter[‡], and M. Verhaegen[‡]

[‡] University of Twente, Faculty TN, SCE Group,

P.O. Box 217, 7500 AE Enschede, The Netherlands, S.K.Kanev@TN.UTwente.NL

[‡] Delft University of Technology, Faculty ITS, CSE Group

P.O. Box 5031, 2600 GA Delft, {B.DeSchutter,M.H.G.Verhaegen}@ITS.TUdelft.NL

Abstract

This paper presents a new iterative approach to probabilistic robust controller design, which is an alternative to the recently proposed Subgradient Iteration Algorithm (SIA). In its original version [12] the SIA possesses the useful property of guaranteed convergence in a finite number of iterations, but requires that the radius of a non-empty ball contained in the solution set is known *a-priori*. This rather restrictive assumption was later on released in [3], but only at the expense of an increased number of iterations. The approach in this paper does also not require the knowledge of such a radius, and offers a significant improvement even over the original SIA in terms of the maximum number of possible correction steps that can be executed before a feasible solution is reached. Given an initial ellipsoid that contains the solution set, the approach iteratively generates a sequence of ellipsoids with decreasing volumes, all containing the solution set. A method for finding an initial ellipsoid containing the solution set is also proposed. The approach is illustrated on a real-life diesel actuator benchmark model.

keywords: probabilistic robustness, robust LMIs, robust control.

1 Introduction

The Subgradient Iteration Algorithm was recently proposed in the literature for probabilistic design of LQ regulators [12]. It was developed in parallel with [3], which deals with general robust LMIs. The main advantage of this approach over the existing deterministic approaches to robust controller design is that it can handle very general uncertainty structures, where the uncertainty can enter the system in any (possibly non-linear) fashion. Additionally, this approach does not solve simultaneously a number of LMIs, whose dimension grows exponentially with the number of uncertain parameters, but rather “solves” one LMI at each iteration.

This turns out to be a very powerful feature when one observes that even for a small number of real uncertain parameters most of the existing LMI solvers will be unable to handle the resulting number of LMIs. For an overview of the literature on probabilistic design the reader is referred to [12, 3, 10, 9, 7, 13, 14, 11, 8], and the references therein.

While enjoying these nice properties, the major drawback of the SIA is that the radius of a ball contained in the solution set is required to be known *a-priori*. This radius is used at each iteration of the SIA to compute the size of the step that will be made in the direction of the anti-gradient of a suitably defined function. As shown later on, not knowing such a radius r may result in the SIA failing to find a feasible solution. Knowledge of r , on the other hand, guarantees that the algorithm will terminate at a feasible solution in a finite number of iterations with probability one, provided that the solution set has a non-empty interior [12]. While a modification of the SIA was proposed in [3] that no longer necessitates the knowledge of r , this was only achieved at the expense of a significant increase in the number of iterations performed. This paper provides an alternative method that offers a strong improvement in the convergence speed even over the original SIA, while at the same time assumes no knowledge of r .

The approach proposed in this paper is based on the *Ellipsoid Algorithm* (EA) [2], and can be used for finding exact or approximate solutions to LMI feasibility problems, like those arising from many (robust) controller/filter design problems. The uncertainty Δ is assumed to be bounded in the structured uncertainty set Δ , and to be coupled with a probability density function $f_{\Delta}(\Delta)$. It is further assumed that it is possible to generate samples of Δ according to $f_{\Delta}(\Delta)$. The interested reader is referred to [5, 6, 4] for more details on the available algorithms for uncertainty generation. Then at each iteration of EA two steps are performed. At the first step a random uncertainty sample $\Delta^{(i)} \in \Delta$ is generated according to the given probability density function $f_{\Delta}(\Delta)$. With this generated uncertainty a suitably defined convex function is parametrized so

¹This work is sponsored by the Dutch Technology Foundation (STW) under project number DEL4506.

that at the second step of the algorithm an ellipsoid is computed, in which the solution set is guaranteed to lie. The EA thus produces a sequence of ellipsoids with decreasing volumes, all containing the solution set. Provided that the solution set has a non-empty interior, it will be established that this algorithm converges to a feasible solution in a finite number of iterations with probability one. It is also shown that even if the solution set has a zero volume, the EA converges to the solution set when the iteration number tends to infinity – a property not possessed by the SIA. In addition, a method is presented for obtaining an initial ellipsoid that contains the solution set.

The remaining part of the paper is organized as follows. In the next Section the problem is formulated, and the SIA is summarized. In Section 3 the EA is developed, its convergence is established and compared to the convergence of the SIA. In Section 4 a method for obtaining an initial ellipsoid containing the solution set is presented. The complete EA method is illustrated in Section 5 on the design of a robust \mathcal{H}_2 state-feedback controller for a benchmark model, taken from [1]. Finally, Section 6 concludes the paper.

2 Introduction to the Problem

2.1 Notation and Problem Formulation

The notation used in this paper is standard besides that $\|\cdot\|$ denotes any matrix norm. The symbols \bullet in LMIs will be used to indicate entries that can be implied by symmetry.

Denoting the cone of symmetric non-negative definite matrices as $\mathcal{C}^+ \triangleq \{A : A = A^T > 0\}$, we define the projection Π^+ that maps symmetric matrices to the cone of symmetric non-negative definite matrices as $\Pi^+A \triangleq \arg \min_{X \in \mathcal{C}^+} \|A - X\|_F$. This projection can be found explicitly as follows. Let the eigenvalue decomposition of the symmetric matrix A be $A = U\Lambda U^T$, with $\Lambda = \text{diag}\{\lambda_1, \dots, \lambda_n\}$. Then it can be shown (see [12]) that $\Pi^+A = U\text{diag}\{\lambda_1^+, \dots, \lambda_n^+\}U^T$, with $\lambda_i^+ = \max(0, \lambda_i)$, $i = 1, \dots, n$.

In this paper we consider an uncertain system $G_\Delta(\sigma)$, where the symbol σ represents the s -operator (*i.e.* the time-derivative operator) for continuous-time systems, and the z -operator (*i.e.* the shift operator) for discrete-time systems. The uncertainty Δ is characterized by an uncertainty set Δ and a probability distribution over this uncertainty set f_Δ .

Many controller (and filter) design problems can be represented in terms of LMIs of the form [2]

Control Problem: Find $x \in \mathcal{X} \subseteq \mathbb{R}^N$ such that $U(x, \Delta) \leq 0$, for all $\Delta \in \Delta$,

where $U(x, \Delta) = U^T(x, \Delta)$ is affine in x , and where the set \mathcal{X} is assumed to be convex. The controller is then parametrized by any solution x^* . Such a controller is called *robust* whenever the uncertainty set Δ has more than one element. The set of all feasible solutions to the control problem is called the *feasibility (solution) set*, and is denoted as

$$\mathcal{S} \triangleq \{x \in \mathcal{X} : U(x, \Delta) \leq 0, \forall \Delta \in \Delta\}. \quad (1)$$

Similarly to [12, 3] we define the following *cost function*

$$v(x, \Delta) \triangleq \|\Pi^+[U(x, \Delta)]\| \geq 0, \quad (2)$$

which is such that

$$\{x \in \mathcal{X} : v(x, \Delta) = 0, \forall \Delta \in \Delta\} \equiv \mathcal{S}.$$

We can thus reformulate the initial control problem in the form of an optimization problem as follows:

$$\text{Find } x^* \in \mathcal{S}. \quad (3)$$

Since $U(x, \Delta)$ is affine in x , it can be represented in the form

$$U(x, \Delta) = U_0(\Delta) + \sum_{i=1}^N U_i(\Delta)x_i,$$

where x_i is the i -th element of the vector x . Then we have the following result, taken from [3].

Lemma 1 *The function $v(x, \Delta)$, defined in Equation (2), is convex in $x \in \mathcal{X}$, and has a subgradient given by*

$$\nabla v(x, \Delta) = \begin{cases} \begin{bmatrix} \text{Tr}(U_1(\Delta)W(x, \Delta)) \\ \vdots \\ \text{Tr}(U_N(\Delta)W(x, \Delta)) \end{bmatrix}, & \text{if } v(x, \Delta) \neq 0, \\ 0, & \text{if } v(x, \Delta) = 0, \end{cases}$$

where $W(x, \Delta) \triangleq \Pi^+[U(x, \Delta)]/v(x, \Delta)$.

2.2 The Subgradient Iteration Algorithm (SIA)

In this Subsection some facts about the recently proposed SIA are summarized. In the original version of the SIA [12], the following assumption is imposed.

Assumption 1 (Strong Feasibility Condition) *A scalar $r > 0$ is known for which there exists $x^* \in \mathcal{X}$ such that $\{x \in \mathcal{X} : \|x - x^*\| \leq r\} \subseteq \mathcal{S}$.*

Assumption 1 implies that the solution set \mathcal{S} has a non-empty interior, and that a radius r of a ball contained in \mathcal{S} is known. This is often a rather restrictive assumption due to the fact that usually no *a-priori* information about the solution set is available.

Define the operator $\Pi_{\mathcal{X}} : \mathbb{R}^N \mapsto \mathcal{X}$ as

$$\Pi_{\mathcal{X}} x \triangleq \arg \min_{y \in \mathcal{X}} \|x - y\|.$$

Then the $(i + 1)$ -th iteration of the SIA is summarized as follows.

Algorithm 1 (Subgradient Iteration Algorithm)

Given $x^{(i)}$ and $0 < \eta < 2$, perform the following steps.

Step 1: Generate a random sample $\Delta^{(i)}$ with probability distribution f_{Δ} .

Step 2: Select the step-size

$$\mu_k = \begin{cases} \eta \frac{v(x^{(i)}, \Delta^{(i)}) + r \|\nabla v(x^{(i)}, \Delta^{(i)})\|}{\|\nabla v(x^{(i)}, \Delta^{(i)})\|^2} & \text{if } v(x^{(i)}, \Delta^{(i)}) \neq 0 \\ 0 & \text{if } v(x^{(i)}, \Delta^{(i)}) = 0, \end{cases}$$

and compute

$$x^{(i+1)} = \Pi_{\mathcal{X}}[x^{(i)} - \mu_k \nabla v(x^{(i)}, \Delta^{(i)})]. \quad (4)$$

The following technical assumption needs to be additionally imposed

Assumption 2 For any $x^{(i)} \notin \mathcal{S}$ there is a non-zero probability to generate a sample $\Delta^{(i)}$ for which $v(x^{(i)}, \Delta^{(i)}) > 0$, i.e. $\text{Prob}(v(x^{(i)}, \Delta) > 0) > 0$.

This assumption is not restrictive and is needed to make sure that for any $x^{(i)} \notin \mathcal{S}$ there is a non-zero probability for a *correction step* to be executed. By correction step it is meant an iteration (4) with $x^{(i+1)} \neq x^{(i)}$.

It is shown in [3] that for any initial condition $x^0 \in \mathcal{X}$, the SIA finds a feasible solution with probability one in a finite number of iterations, provided that Assumptions 1 and 2 hold. It is also shown that the number

$$I_{SIA} = \|x^0 - x^*\|^2 / (r^2 \eta (2 - \eta)) \quad (5)$$

provides an upper bound on the maximum number of correction steps that can be executed.

A serious drawback of the SIA is that Assumption 1 is too restrictive as in most applications r is unknown. The next example demonstrates how the SIA may actually result in divergence in cases that the radius r is inappropriately selected.

Example 1 Consider the discrete-time system

$$\mathcal{M} : x_{k+1} = x_k + u_k. \quad (6)$$

and the cost function

$$J_{LQR}(k) = \sum_{i=1}^{\infty} x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i}.$$

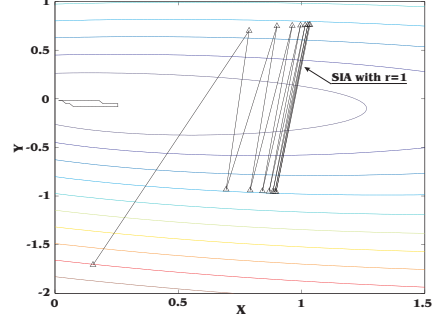


Figure 1: Performance of the Subgradient Iteration Algorithm (SIA) for system \mathcal{M} in Example 1.

Let $X = X^T > 0$ and Y be such that

$$\begin{bmatrix} X & (AX + BY)^T & XQ^{1/2} & Y^T R^{1/2} \\ AX + BY & X & 0 & 0 \\ Q^{1/2} X & 0 & I & 0 \\ R^{1/2} Y & 0 & 0 & I \end{bmatrix} \geq 0.$$

Then it is a fact that the control law $u_k = YX^{-1}x_k$ achieves $J_{LQR}(k) \leq x_k^T X^{-1}x_k$ [2]. Selecting $Q = 1$, $R = 10$, $r = 1$, $X_0 = 0.1545$, $Y_0 = -1.7073$, the SIA does not converge to the solution set, but rather begins to oscillate, as it can be seen from Figure 1. The optimum is represented by the innermost contour in Figure 1 (left). The contours in Figure 1 represent different level sets. A level set $LS(c, \Delta^*)$ of the function $v(x, \Delta^*)$ for a given $\Delta^* \in \Delta$ and $c \geq 0$ is defined as

$$LS(c, \Delta^*) \triangleq \{x \in \mathcal{X} : v(x, \Delta^*) \leq c\}. \quad (7)$$

In [3] it is proposed that, whenever unknown, the radius r in the SIA is replaced by $\varepsilon_{s(i)}$, where $s(i)$ is the number of correction steps performed before the i -th iteration, and $\varepsilon_s > 0$, $\varepsilon_s \rightarrow 0$, and $\sum_{s=0}^{\infty} \varepsilon_s = \infty$. While this modification retains the convergence properties of the SIA, it can significantly increase the number of correction steps, and thus the number of iterations that need to be performed (see Equation (5)).

3 The Ellipsoid Algorithm (EA)

In this section an alternative approach to the SIA is proposed that both requires no knowledge of r , and retains the nice convergence properties of the SIA.

Assume that an initial ellipsoid $E^{(0)}$ that contains the solution set \mathcal{S} is given

$$E^{(0)} = \{x \in \mathcal{X} : (x - x^{(0)})^T P_0^{-1} (x - x^{(0)}) \leq 1\} \supseteq \mathcal{S}$$

with center $x^{(0)} \in \mathcal{X}$ and $P_0 \in \mathbb{R}^{N \times N}$ such that $P_0 = P_0^T > 0$. The problem of finding such an initial ellipsoid

will be discussed in the next section. Define the half-space

$$H^{(0)} \triangleq \{x \in \mathcal{X} : \nabla^T v(x^{(0)}, \Delta)(x - x^{(0)}) \leq 0\}.$$

Due to the convexity of the function $v(x, \Delta)$ we know that $H^{(0)}$ also contains the solution set \mathcal{S} , and therefore $\mathcal{S} \subseteq H^{(0)} \cap E^{(0)}$. We can then construct a new ellipsoid, $E^{(1)}$, as the *minimum volume* ellipsoid such that $E^{(1)} \supseteq H^{(0)} \cap E^{(0)} \supseteq \mathcal{S}$, and such that the volume of $E^{(1)}$ is less than the volume of $E^{(0)}$. This, repeated iteratively, represents the main idea behind the Ellipsoid Algorithm [2]. Suppose at iteration $(i+1)$ we have $x^{(i)} \in \mathcal{X}$ and $P_i = P_i^T > 0$ such that

$$E^{(i)} = \{x \in \mathcal{X} : (x - x^{(i)})^T P_i^{-1} (x - x^{(i)}) \leq 1\} \supseteq \mathcal{S}.$$

The $(i+1)$ -th iteration of the EA is then summarized as follows.

Algorithm 2 (Ellipsoid Algorithm) Given $x^{(i)} \in \mathcal{X} \subseteq \mathbb{R}^N$ and $P_i = P_i^T > 0$, perform the following two steps

Step 1: Generate a random sample $\Delta^{(i)}$ with probability distribution f_Δ .

Step 2: Form the ellipsoid

$$E^{(i+1)} = \{x \in \mathcal{X} : (x - x^{(i+1)})^T P_{i+1}^{-1} (x - x^{(i+1)}) \leq 1\} \supseteq \mathcal{S}.$$

with

$$x^{(i+1)} = x^{(i)} - \frac{1}{N+1} \frac{P_i \nabla v(x^{(i)}, \Delta^{(i)})}{\sqrt{\nabla^T v(x^{(i)}, \Delta^{(i)}) P_i \nabla v(x^{(i)}, \Delta^{(i)})}},$$

$$P_{i+1} = \frac{N^2}{N^2-1} \left(P_i - \frac{2}{N+1} \frac{P_i \nabla v(x^{(i)}, \Delta^{(i)}) \nabla^T v(x^{(i)}, \Delta^{(i)}) P_i^T}{\nabla^T v(x^{(i)}, \Delta^{(i)}) P_i \nabla v(x^{(i)}, \Delta^{(i)})} \right)$$

if $v(x^{(i)}, \Delta^{(i)}) \neq 0$, and $x^{(i+1)} = x^{(i)}$ and $P_{i+1} = P_i$ if $v(x^{(i)}, \Delta^{(i)}) = 0$.

The convergence of the approach is established in the following lemma.

Lemma 2 (Convergence of the EA) Consider the EA, and suppose that Assumption 2 holds. Let

- (i) $\text{vol}(\mathcal{S}) > 0$. Then a feasible solution will be found in a finite number of iterations with probability one.
- (i) $\text{vol}(\mathcal{S}) = 0$. Then $\lim_{i \rightarrow \infty} x^{(i)} = x^* \in \mathcal{S}$ with probability one.

Proof: The EA generates ellipsoids with geometrically decreasing volumes [2], i.e.

$$\text{vol}(E^{(i)}) \leq e^{-\frac{s(i)}{2N}} \text{vol}(E^{(0)}),$$

where $s(i)$ is the number of correction steps performed before iteration i . Thus, under Assumption 2 we can write

$$\lim_{i \rightarrow \infty} \text{vol}(E^{(i)}) = 0. \quad (8)$$

(i) If we now suppose that the solution set \mathcal{S} has a non-empty interior, i.e. $\text{vol}(\mathcal{S}) > 0$, then from Equation (8) and due to the fact that $E^{(i)} \supseteq \mathcal{S}$ for all $i = 0, 1, \dots$, it follows that in a finite number of iterations with probability one the algorithm will terminate at a feasible solution.

(ii) If we now suppose that $\text{vol}(\mathcal{S}) = 0$, then due to the convexity of the function, and due to Equation (8), the algorithm will converge to a point in \mathcal{S} with probability one. ■

Thus, the newly proposed EA converges to a feasible solution even in the case that the set \mathcal{S} has an empty interior, a property not possessed by the SIA.

Finally, similarly to the bound I_{SIA} (5) on the maximum number of correction steps for the SIA, we can derive such an upper bound for the proposed EA.

Lemma 3 Consider the EA, and suppose that Assumption 2 holds. Suppose further that the solution set has a non-empty interior, i.e. $\text{vol}(\mathcal{S}) > 0$. Then the number

$$I_{EA} = 2N \left[\ln \frac{\text{vol}(E^{(0)})}{\text{vol}(\mathcal{S})} \right]_+ \quad (9)$$

is an upper bound on the maximum number of correction steps that can be performed starting from any ellipsoid $E^{(0)} \supseteq \mathcal{S}$, where $[a]_+$, $a \in \mathbb{R}$, denotes the minimum integer number larger than or equal to a .

Proof: It is shown in [2] that

$$\text{vol}(E^{(i)}) \leq e^{-\frac{s(i)}{2N}} \text{vol}(E^{(0)}).$$

Since the volume of the consecutive ellipsoids tends to zero, and since $\text{vol}(\mathcal{S}) > 0$, there exists a number I_{EA} such that

$$e^{-\frac{s(i)}{2N}} \text{vol}(E^{(0)}) \leq \text{vol}(\mathcal{S}), \quad \forall s(i) \geq I_{EA}.$$

Therefore, we could obtain the number I_{EA} from the following relation

$$\frac{\text{vol}(\mathcal{S})}{\text{vol}(E^{(0)})} \geq e^{-\frac{s(i)}{2N}} \iff s(i) \geq I_{EA}$$

Now, by taking the natural logarithm on both sides one obtains

$$\ln \frac{\text{vol}(\mathcal{S})}{\text{vol}(E^{(0)})} \geq -\frac{s(i)}{2N} \iff s(i) \geq I_{EA}$$

or

$$s(i) \geq 2N \ln \frac{\text{vol}(E^{(0)})}{\text{vol}(\mathcal{S})} \iff s(i) \geq I_{EA}$$

Therefore, Equation (9) is proven. ■

We would like to point out that usually $I_{EA} \ll I_{SIA}$. This is demonstrated in the following example.

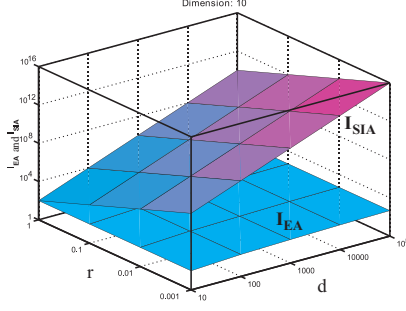


Figure 2: Comparison between the upper bounds I_{EA} and I_{SIA} for the algorithms SIA and EA.

Example 2 (Comparison between I_{EA} and I_{SIA})

Let us suppose that the dimension of our vector of unknowns is 10 (i.e. $N = 10$), and that the solution set is a ball of radius 1.1 and center $x^* \in \mathbb{R}^{10}$, i.e. $\mathcal{S} = \{x \in \mathbb{R}^{10} : \|x - x^*\| \leq 1.1\}$. To make a fair comparison between the SIA and the newly proposed EA we assume that the initial condition $x^{(0)}$ for the SIA is at a distance d from the center of \mathcal{S} , i.e. $\|x^{(0)} - x^*\| = d$, and that the initial ellipsoid for the EA is a ball of radius d . Since for SIA the number r in Assumption 1 should be known, we will make several experiments with $r = \{0.001, 0.01, 0.1, 1\}$. For these values of r , and for $d = \{10, 10^2, 10^3, 10^4, 10^5\}$ the two upper bounds I_{EA} and I_{SIA} on the maximum numbers of possible correction steps for the two algorithms were computed. Figure 2 represents the results (note that all the three axes are in logarithmic scale). Clearly, $I_{EA} \ll I_{SIA}$.

4 Finding an Initial Ellipsoid $E^{(0)}$

This section presents a method for obtaining an initial ellipsoid that contains the solution set. Using the level set definition given in Example 1 and due to the convexity of the function $v(x, \Delta)$, it is clear that the solution set \mathcal{S} is contained in any level set $LS(c, \Delta)$ for any $c \geq 0$ and any $\Delta \in \Delta$. In fact, we have that

$$\mathcal{S} = \bigcap_{\Delta \in \Delta} LS(0, \Delta).$$

Thus, $LS(c, \Delta) \supseteq LS(0, \Delta) \supseteq \mathcal{S}$ for any $c \geq 0$ and any $\Delta \in \Delta$. Therefore, \mathcal{S} is also contained in $LS(0, 0)$, i.e. $\mathcal{S} \subseteq LS(0, 0)$. The idea is then to find an initial ellipsoid that contains the level set $LS(0, 0)$, under the assumption that this is a bounded set¹. To this end we will first bound the set $LS(0, 0)$ with a rectangular parallelepiped, and then we build an ellipsoid around

¹Note that unbounded solutions are clearly of no practical interest here since these would lead to physically unrealizable controllers/filters. It thus might be reasonable for some applications to put hard bounds on the entries of x .

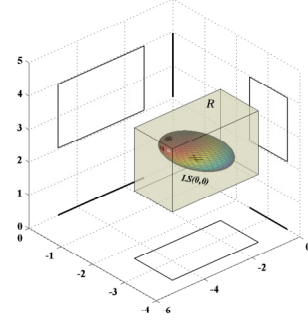


Figure 3: Initial ellipsoid computation.

it as shown in Figure 3, which we will use as an initial ellipsoid to start the EA. In order to find a bounding rectangular parallelepiped, we need to find solutions to the following constrained optimization problems

$$\begin{aligned} \bar{x}_i &= \max_{x \in \mathcal{X}} x_i, \text{ subject to } x \in LS(0, 0), \quad i = 1, 2, \dots, N, \\ \underline{x}_i &= \min_{x \in \mathcal{X}} x_i, \text{ subject to } x \in LS(0, 0), \quad i = 1, 2, \dots, N, \end{aligned}$$

These can be rewritten as LMI problems by noting that $\{x \in LS(0, 0)\} \equiv \{x \in \mathcal{X} : v(x, 0) = 0\} \equiv \{x \in \mathcal{X} : U(x, 0) \leq 0\}$.

As a result, the following algorithm is proposed for fast initial ellipsoid selection.

Algorithm 3 (Initial Ellipsoid Computation)

(a) Find solutions to the LMI problems

$$\begin{aligned} \bar{x}_i &= \max_{x \in \mathcal{X}} e_i^T x, \text{ subject to } U(x, 0) \leq 0, \quad i = 1, 2, \dots, N, \\ \underline{x}_i &= \min_{x \in \mathcal{X}} e_i^T x, \text{ subject to } U(x, 0) \leq 0, \quad i = 1, 2, \dots, N, \end{aligned}$$

with $e_i = [0_{1 \times (i-1)}, 1, 0_{1 \times (n-i)}]^T$.

(b) Take $\bar{x} = [\bar{x}_1, \dots, \bar{x}_N]^T$ and $\underline{x} = [\underline{x}_1, \dots, \underline{x}_N]^T$, and define the box

$$R = \{x : \underline{x} \leq x \leq \bar{x}\} \supseteq LS(0, 0) \supseteq \mathcal{S}.$$

(c) Next, find an ellipsoid that encircles the box R . This can easily be done by first finding an ellipsoid inside R and then stretching it to embrace R . The ellipsoid

$$E_{in} = \{x \in \mathcal{X} : (x - x_c)^T P^{-1} (x - x_c) \leq 1\}$$

with $x_c = \frac{1}{2}(\bar{x} + \underline{x})$ and $P = \text{diag} \frac{1}{2}(\bar{x} - \underline{x})^2$ is inside R . By defining $A = P^{-1/2}$ and $b = -P^{-1/2}x_c$, this ellipsoid can be equivalently represented as

$$E_{in} = \{x \in \mathcal{X} : \|Ax + b\|_2^2 \leq 1\}.$$

(d) Stretching the ellipsoid E_{in} by α^2 with $\alpha > 1$ results in

$$E_{out} = \{x \in \mathcal{X} : \alpha^{-2} \|Ax + b\|_2^2 \leq 1\},$$

which we need to be such that it contains both \bar{x} and \underline{x} . Therefore we take $\alpha = \max_{x \in \{\bar{x}, \underline{x}\}} \|Ax + b\|_2$. The initial ellipsoid can then be taken as

$$E_0 = \{x \in \mathcal{X} : (x - x_c)^T (\alpha^2 P)^{-1} (x - x_c) \leq 1\}.$$

5 An Illustrative Example

Next, we present an example illustrating the probabilistic approach developed in this paper. We consider the problem of designing a robust \mathcal{H}_2 state-feedback controller for a model, representing a real-life diesel actuator benchmark system, taken from [1]. The model, which will not be reported here due to lack of space, has four real uncertain parameters. The goal is to design a robust state-feedback controller for the uncertain system that achieves an upper bound $\gamma_{UB} = 1$ for the worst case \mathcal{H}_2 -norm of the closed-loop system. This problem can be represented as the following LMI feasibility problem: Find Q , R , and L such that

$$\begin{aligned} & \text{trace}(R) < 1 \\ & \begin{bmatrix} R & C_{z,i}Q + D_{zu,i}L \\ \bullet & Q \\ -A_iQ - QA_i^T - B_{u,i}L - L^TB_{u,i}^T & B_{\xi,i} \\ \bullet & I \end{bmatrix} > 0 \end{aligned}$$

Then $F = LQ^{-1}$ is the desired state-feedback gain matrix.

Application of the proposed approach resulted in the state-feedback gain matrix

$$F = \begin{bmatrix} -0.81508 & -0.64339 & -3.2121 \times 10^{-2} \end{bmatrix}.$$

This solution was found by the EA method in less than 100 iterations. Starting from the same initial conditions, the SIA was terminated after 500 iterations having found no feasible solution (it was run for $r = 1$, $r = 0.1$, and $r = 0.01$).

6 Conclusions

In this paper a new probabilistic approach was proposed to robust controller design via LMIs, based on the Ellipsoid Algorithm. It features a number of advantages over the probabilistic Subgradient Iteration Algorithm, recently proposed in [12, 3]. In its original version [12] the SIA relies on the restrictive assumption that a radius r of a ball contained in the solution set is known. This assumption was later on released in a modification of the SIA in [3], but only at the expense of reduced computational speed. The approach in this paper is an alternative to this modified SIA that does not rely on knowledge of r , while at the same time enjoying increased computational speed compared to the original SIA. It was shown that it converges to a feasible solution in a finite number of iterations with probability one provided that the solution set has a non-empty interior. The approach was demonstrated on the design of a robust \mathcal{H}_2 state-feedback controller for a benchmark example.

References

- [1] M. Blanke, S.A. Bøgh, R.B. Jørgenson, R.J. Patton, Fault Detection for a Diesel Engine Actuator – A Benchmark for FDI, *Control Engineering Practice*, 3(12), 1995, pp. 1731-1740.
- [2] S. Boyd, L. Ghaoui, E. Feron, V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, volume 15 of Studies in Applied Mathematics, SIAM, Philadelphia, PA, June 1994.
- [3] G. Calafiore, B. Polyak, Stochastic Algorithms for Exact and Approximate Feasibility of Robust LMIs, *IEEE Transactions on Automatic Control*, 46(11), 2001, pp. 1755-1759.
- [4] G. Calafiore, F. Dabbene, R. Tempo, Randomized Algorithms for Probabilistic Robustness with Structured Uncertainty, *Proceedings of the 38th IEEE Conference on Decision and Control CDC 1999*, Phoenix, Arizona, pp. 528-533.
- [5] G. Calafiore, F. Dabbene, R. Tempo, Radial and Uniform Distributions in Vector and Matrix Spaces for Probabilistic Robustness, In *Topics in Control and its Applications* (Eds. D.E. Miller and L. Qiu), Springer-Verlag, London, pp. 17-31, 1999.
- [6] G. Calafiore, F. Dabbene, R. Tempo, Randomized Algorithms for Probabilistic Robustness with Real and Complex Structured Uncertainty, *IEEE Transactions on Automatic Control*, 45(12), 2000, pp. 2218-2235.
- [7] G. Calafiore, F. Dabbene, R. Tempo, Randomized Algorithms for Reduced Order \mathcal{H}_∞ Controller Design, *Proceedings of the American Control Conference ACC 2000*, Chicago, Illinois, pp. 3837-3839.
- [8] X. Chen, K. Zhou, Order Statistics and Probabilistic Robust Control, *Systems & Control Letters*, 35(3), 1998, pp. 175-182.
- [9] Y. Fujisaki, F. Dabbene, R. Tempo, Probabilistic Robust Design of LPV Control Systems, *Proceedings of the 40th IEEE Conference on Decision and Control CDC 2001*, Orlando, Florida, pp. 2019-2024.
- [10] Y. Oishi, H. Kimura, Randomized Algorithm to Solve Parameter Dependent Linear Matrix Inequalities and Their Computational Complexity, *Proceedings of the 40th IEEE Conference on Decision and Control CDC 2001*, Orlando, Florida, pp. 2025-2030.
- [11] K. Öhrn, A. Ahlén, M. Sternad, A Probabilistic Approach to Multivariable Robust Filtering and Open-Loop Control, *IEEE Transactions on Automatic Control*, 40(3), 1995, pp. 405-418.
- [12] B. Polyak, R. Tempo, Probabilistic Robust Design with Linear Quadratic Regulators, *Systems & Control Letters*, 43(5), 2001, pp. 343-353.
- [13] R. Stengel, L. Ray, Stochastic Robustness of Linear Time-Invariant Control Systems, *IEEE Transactions on Automatic Control*, 36(1), 1991, pp. 82-87.
- [14] M. Vidyasagar, V. Blondel, Probabilistic Solutions to Some NP-Hard Matrix Problems, *Automatica*, 37(9), 2001, pp. 1397-1405.