Delft University of Technology

Fac. of Information Technology and Systems **Control Systems Engineering**

Technical report CSE02-017

An 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, "An ellipsoid algorithm for probabilistic robust controller design," *Systems & Control Letters*, vol. 49, no. 5, pp. 365–375, Aug. 2003. doi:10.1016/S0167-6911(03)00115-4

Control Systems Engineering
Faculty of Information Technology and Systems
Delft University of Technology
Delft, The Netherlands
Current URL: https://www.dcsc.tudelft.nl

An Ellipsoid Algorithm for Probabilistic Robust Controller Design

S. Kanev [a,1], B. De Schutter [b] and M. Verhaegen [c]

[a] *University of Twente, Faculty of Applied Physics, Systems and Control Engineering Group, P.O. Box 217, 7500 AE Enschede, The Netherlands, S.Kanev@ITS.TUDelft.NL*

[b] *Delft University of Technology, Faculty of Information Technology & Systems, Control Systems Engineering Group, P.O. Box 5031, 2600 GA Delft, The Netherlands, B.DeSchutter@ITS.TUDelft.NL*

[c] *Delft University of Technology, Faculty of Information Technology & Systems, Control Systems Engineering Group, P.O. Box 5031, 2600 GA Delft, The Netherlands, M.H.G.Verhaegen@ITS.TUDelft.NL*

Abstract

In this paper a new iterative approach to probabilistic robust controller design is presented, which is applicable to any robust controller/filter design problem that can be represented as an LMI feasibility problem. Recently, a probabilistic Subgradient Iteration algorithm was proposed for solving LMIs. It transforms the initial feasibility problem to an equivalent convex optimization problem, which is subsequently solved by means of an iterative algorithm. While this algorithm always converges to a feasible solution in a finite number of iterations, it requires that the radius of a non-empty ball contained into the solution set is known *a-priori*. This rather restrictive assumption is released in this paper, while retaining the convergence property. Given an initial ellipsoid that contains the solution set, the approach proposed here iteratively generates a sequence of ellipsoids with decreasing volumes, all containing the solution set. At each iteration a random uncertainty sample is generated with a specified probability density, which parametrizes an LMI. For this LMI the next minimum-volume ellipsoid that contains the solution set is computed. An upper bound on the maximum number of possible correction steps, that can be performed by the algorithm before finding a feasible solution, is derived. A method for finding an initial ellipsoid containing the solution set, which is necessary for initialization of the optimization, is also given. The proposed approach is illustrated on a real-life diesel actuator benchmark model with real parametric uncertainty, for which a $\mathcal{H}_2$ robust state-feedback controller is designed.

Key words: Probabilistic design, Randomized algorithms, Robust LMIs, Robust control, Ellipsoid algorithm.

Preprint submitted to Elsevier

1 Introduction

Recently, a new approach for probabilistic design of LQ regulators was proposed in the literature [12], to which we will refer to as the *Subgradient Iteration Algorithm* (SIA), which was later on extended to deal with general robust LMIs [3]. 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. In addition to that, this approach does not need to 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 ten 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 [3,12–16,10,5,6], 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 (the set of all feasible solutions to the problem) is required to be known *a-priori*. This radius is used at each iteration of the SIA to compute the size of the step which will be made in the direction of the anti-gradient of a suitably defined convex function. It will be shown later in this paper that not knowing such a radius r can result in the SIA failing to find a feasible solution. Knowing r, on the other hand, guarantees that the algorithm will terminate in a feasible solution in a finite number of iterations with probability one, provided that the solution set has a non-empty interior [12,3]. The purpose of this paper is to develop a new probabilistic approach that no longer necessitates the knowledge of r, while keeping the above-mentioned advantages and the convergence property of SIA.

To circumvent the lack of knowledge of r, it is proposed in [11] that one can substitute this number with a sequence $\{\epsilon_s\}$ such that $\epsilon_s > 0$, $\epsilon \to 0$ and $\sum_{s=0}^{\infty} \epsilon_s = \infty$. While this indeed releases the assumption that the radius r is known, it increases the number of iterations necessary to arrive at a feasible solution. In addition to that the choice of an appropriate sequence $\{\epsilon_s\}$ remains an open question.

An interesting result concerning the algorithm in [3] appeared recently in [8], where it is shown that the expected time to achieve a solution is infinite. In [8] the authors also propose a slight modification of the approach from [3] that results in an algorithm with finite expected achievement time. Yet, this modified algorithm suffers from the "curse of dimensionality", i.e. the

[1] This work is sponsored by the Dutch Technology Foundation (STW) under project number DEL.4506

expected achievement time grows (faster than) exponentially with the number of uncertain parameters.

The approach proposed in this paper is based on the *Ellipsoid Algorithm* (EA). The algorithm can be used for finding exact or approximate solutions to LMI optimization problems, like those arising from many (robust) controller and filter design problems. The uncertainty Δ is assumed to be bounded in the structured uncertainty set $\mathbf{\Delta}$, and to be coupled with a probability density function $f_{\mathbf{\Delta}}(\Delta)$. It is further assumed that it is possible to generate samples of Δ according to $f_{\mathbf{\Delta}}(\Delta)$. The interested reader is referred to [4] for more details on the available algorithms for uncertainty generation. Then, similarly to the SIA, at each iteration of the EA two steps are performed. In the first step a random uncertainty sample $\Delta^{(i)} \in \mathbf{\Delta}$ is generated according to the given probability density function $f_{\mathbf{\Delta}}(\Delta)$. With this generated uncertainty a suitably defined convex function is parametrized so 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. Using some existing facts, and 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. To initialize the algorithm, a method is presented for obtaining an initial ellipsoid that contains the solution set. 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.

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 and its convergence is established. In Section §4 a possible method for finding 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 real-life diesel actuator 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 the paper is as follows. I_n denotes the identity matrix of dimension $n \times n$, $I_{n\times m}$ is a matrix of dimension $n \times m$ with ones on its main diagonal. The dimensions will often be omitted in cases where they can be implied from the context. $\|.\|$ denotes any matrix norm. $A > 0$ ($A \geq 0$) means that A is positive definite (positive semi-definite). We also introduce the

notation $\|x\|_Q^2 \doteq x^T Q x$ for $x \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ with $Q \geq 0$, which should not be mistaken with the standard notation for the vector p-norm ($\|x\|_p$). A vector of dimension n with all elements equal to zero will be denoted as $\mathbf{0}_n$.

Let the space of all symmetric n-by-n matrices be denoted as $\mathcal{S}_n$, and let $\mathcal{C}_n^+$ denote the cone of symmetric non-negative definite n-by-n matrices [12]. Then we define the projection $\mathbf{\Pi}^+ : \mathcal{S}_n \to \mathcal{C}_n^+$ in the following way:

$$\mathbf{\Pi}^+ A \doteq \arg \min_{X \in \mathcal{C}_n^+} \|A - X\|_F.$$

This projection can be found explicitly as follows. For a matrix $A \in \mathcal{S}_n$ the eigenvalue decomposition exists and has the following form $A = U\Lambda U^T$, where U is an orthogonal matrix containing the eigenvectors of A, and Λ is a diagonal matrix with the eigenvalues $\lambda_i,\ i = 1, \ldots, n$, of A appearing on its diagonal, i.e. $\Lambda = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$. Then it can be shown that (see [12])

$$\mathbf{\Pi}^+ A = U \mathrm{diag}\{\lambda_1^+, \ldots, \lambda_n^+\} U^T,$$

with $\lambda_i^+ = \max(0, \lambda_i)$, $i = 1, \ldots, n$.

In this paper we consider the following uncertain transfer function

$$G_\Delta(\sigma) : \begin{bmatrix} u \\ \xi \end{bmatrix} \mapsto \begin{bmatrix} z \\ y \end{bmatrix},$$

defined as

$$G_\Delta(\sigma) = \begin{bmatrix} C_z^\Delta \\ C_y^\Delta \end{bmatrix} (\sigma I_n + A^\Delta)^{-1} \begin{bmatrix} B_u^\Delta & B_\xi^\Delta \end{bmatrix} + \begin{bmatrix} D_{zu}^\Delta & D_{z\xi}^\Delta \\ D_{yu}^\Delta & D_{y\xi}^\Delta \end{bmatrix}. \tag{1}$$

where $A^\Delta \in \mathbb{R}^{n\times n}$, $B_u^\Delta \in \mathbb{R}^{n\times m}$, $B_\xi^\Delta \in \mathbb{R}^{n\times n_\xi}$, $C_z^\Delta \in \mathbb{R}^{n_z\times n}$, $C_y^\Delta \in \mathbb{R}^{p\times n}$, $D_{zu}^\Delta \in \mathbb{R}^{n_z\times m}$, $D_{z\xi}^\Delta \in \mathbb{R}^{n_z\times n_\xi}$, $D_{yu}^\Delta \in \mathbb{R}^{p\times m}$, $D_{y\xi}^\Delta \in \mathbb{R}^{p\times n_\xi}$, $u \in \mathbb{R}^m$ is the control action, $y \in \mathbb{R}^p$ is the measured output, $z \in \mathbb{R}^{n_z}$ is the controlled output of the system, and $\xi \in \mathbb{R}^{n_\xi}$ is the disturbance to the system, and 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 $\mathbf{\Delta}$ and a probability distribution over this uncertainty set $f_{\mathbf{\Delta}}$.

Many controller (and filter) design problems are known to be representable in

terms of LMIs [2]

Control Problem: Find a feasible solution to the LMI

$$U(x, \Delta) \leq 0, \ x \in \mathcal{X} \subseteq \mathbb{R}^N, \ \text{for all } \Delta \in \mathbf{\Delta},$$

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 $\mathbf{\Delta}$ has more than one element.

The set of all feasible solutions to the control problem is called the *solution set*, and is denoted as

$$\mathcal{S} \doteq \{x \in \mathcal{X} : \ U(x, \Delta) \leq 0, \ \forall \Delta \in \mathbf{\Delta}\}. \tag{2}$$

It is assumed throughout this paper that the solution set $\mathcal{S}$ is nonempty.

The goal is the development of an iterative algorithm capable of finding a solution to the control problem defined above. To this end the following *cost function* is defined

$$v(x, \Delta) \doteq \|\mathbf{\Pi}^+[U(x, \Delta)]\| \geq 0, \tag{3}$$

which is such that $v(x, \Delta) = 0$ for all $\Delta \in \mathbf{\Delta}$ if and only if $x \in \mathcal{S}$.

Thus, the initial problem is reformulated to the following optimization problem

$$x^* = \arg \min_{x \in \mathcal{X}} \sup_{\Delta \in \mathbf{\Delta}} v(x, \Delta). \tag{4}$$

Note that x^* is such that $v(x^*, \Delta) = 0$ for every $\Delta \in \mathbf{\Delta}$ due to the assumption that $\mathcal{S}$ is nonempty.

In [3] it is shown that the function $v(x, \Delta)$ is convex in x and a subgradient of $v(x, \Delta)$, denoted here as $\nabla v(x, \Delta)$, is derived.

2.2 The Subgradient Iteration Algorithm (SIA)

For finding a feasible solution to the optimization problem (4), an algorithm was proposed in [3]. It originated in [12], where it was developed specifically for the design of a state-feedback LQ regulator. We will refer to this algorithm

as the *Subgradient Iteration Algorithm* due to the fact that it is based on subgradient iterations.

Define the operator $\mathbf{\Pi}_{\mathcal{X}} : \mathbb{R}^N \mapsto \mathcal{X}$ as follows

$$\mathbf{\Pi}_{\mathcal{X}} x \doteq \arg\min_{y \in \mathcal{X}} \|x - y\|.$$

Further, the following assumption is imposed for the SIA.

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 is a rather restrictive assumption due to the fact that usually no *a-priori* information about the solution set is available. This Assumption will be released in the next Section where the newly proposed algorithm is presented.

The SIA is then summarized as follows (see [12,3] for more details).

Algorithm 1 (Subgradient Iteration Algorithm: iteration $i+1$) *Given $x^{(i)}$ and $0 < \eta < 2$, perform the following steps.*

Step 1. *Generate a random sample $\Delta^{(i)} \in \mathbf{\Delta}$ with probability distribution f_Δ.*
Step 2. *Select the step-size*

$$\mu_i = \begin{cases} \eta \frac{v(x^{(i)}, \Delta^{(i)}) + r\|\nabla v(x^{(i)}, \Delta^{(i)})\|}{\|\nabla v(x^{(i)}, \Delta^{(i)})\|^2} & \text{if } v(x^{(i)}, \Delta^{(i)}) \neq 0 \\ 0 & \text{if } v(x^{(i)}, \Delta^{(i)}) = 0, \end{cases} \tag{5}$$

and compute

$$x^{(i+1)} = \mathbf{\Pi}_{\mathcal{X}}[x^{(i)} - \mu_i \nabla v(x^{(i)}, \Delta^{(i)})]. \tag{6}$$

As an initial condition $x^{(0)}$ to the algorithm can be selected any element of the set $\mathcal{X}$. As a stopping criterion one may, for instance, select the condition that for a given number of iterations L (usually $L \gg 1$) the step-size $\mu_{i-k} = 0$ (or equivalently $v(x^{(i-k)}, \Delta^{(i-k)}) = 0$) for $k = 0, 1, \ldots, L$. A "weaker" stopping condition could be that the vector $x^{(i)}$ did not change significantly in the last L iterations. Once the algorithm has terminated, a Monte-Carlo simulation could be performed to estimate the empirical probability of robust feasibility [3]. Whenever the obtained probability is unsatisfactory, the number L can be increased and the algorithm can be continued until a better solution (achieving higher empirical probability of robust feasibility) is found.

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.*

$$\mathbf{Prob}(v(x^{(i)}, \Delta^{(i)}) > 0) > 0.$$

This assumption is not restrictive and needs to hold also for the algorithm, proposed in the next Section. The Assumption 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 (6) 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

$$I_{SIA} = \|x^{(0)} - x^*\|^2/(r^2\eta(2-\eta)) \tag{7}$$

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

Although there are a lot of applications for which the subgradient algorithm performs well, in general it possesses the weakness that Assumption 1 is too restrictive, i.e. the number r is not known. As it is demonstrated below, if it is selected not small enough, so that the condition in Assumption 1 does not hold, then Algorithm SIA results in an oscillatory sequence $\{x^{(i)}\}_{i=1,2,...}$ that actually *diverges* from the solution set. On the other hand, if r is selected too small to make sure that Assumption 1 is satisfied, then the convergence rate of the algorithm can drastically slow down since the maximum number of correction steps is reversely proportional to r^2. To experimentally illustrate this discussion we consider the following example

Example 1 *Consider the discrete-time system*

$$\mathcal{M}: \ x_{k+1} = x_k + u_k, \tag{8}$$

and the following standard LQ cost function is minimized

$$J_{LQR} = \sum_{i=1}^{\infty} \|x_{k+i}\|_Q^2 + \|u_{k+i}\|_R^2.$$

It is shown in [9] that the control action $u_k = Fx_k = YX^{-1}x_k$ achieves an upper bound of $x_k^T X^{-1} x_k$ on the cost function if and only if $X = X^T > 0$ and

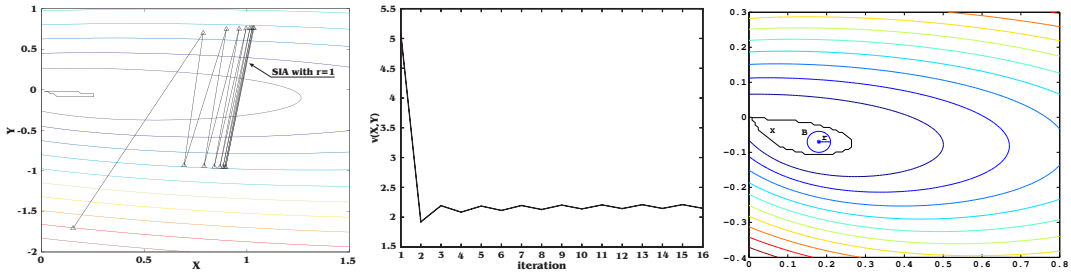

Fig. 1. Performance of the Subgradient Iteration Algorithm (SIA) for system $\mathcal{M}$: (left) level curves of $v([X\ \ Y]^T)$ together with a plot of the sequence $\{[X^{(i)},\ Y^{(i)}]^T\}_{i=1}^{16}$, (middle) plot of $v([X^{(i)},\ Y^{(i)}]^T)$ versus the iteration number i, (right) a zoom on the solution set.

Y *are such that*

$$\begin{bmatrix} X & (AX+BY)^T & XQ^{1/2} & Y^TR^{1/2} \\ AX+BY & X & 0 & 0 \\ Q^{1/2}X & 0 & I & 0 \\ R^{1/2}Y & 0 & 0 & I \end{bmatrix} \geq 0. \tag{9}$$

By (randomly) selecting $Q = 1$, $R = 10$, $r = 1$, $\eta = 1$, $X_0 = 0.1545$, $Y_0 = -1.7073$, *the subgradient iteration algorithm does not converge to the solution set, but rather begins to oscillate, as it can be seen from Figure 1. The feasibility set is represented by the innermost contour in Figure 1 (left). The contours in Figure 1 represent different level sets. A level set* $LS(c, \Delta^*)$ *for the function* $v(x, \Delta)$ *for a given* $\Delta^* \in \mathbf{\Delta}$ *and for a given positive number* c *is defined as*

$$LS(c, \Delta^*) \doteq \{x \in \mathcal{X} :\ v(x, \Delta^*) \leq c\}. \tag{10}$$

The reason for these oscillations is that there exists no ball of radius $r = 1$ *inside the solution set (see Figure 1 (right)). Clearly, for this trivial example one can obtain convergence by simply reducing* r *a bit (for instance, taking* $r = 0.5$ *results in convergence to a solution in six iterations), but in general for larger systems of LMIs simple trial-and-error method with different values of the radius* r *may not be the best option.*

The approach proposed in this paper is based on the *Ellipsoid Algorithm* (EA). Similarly to the SIA method, at each iteration of the EA two steps are performed. In the first step a random uncertainty sample $\Delta^{(i)} \in \mathbf{\Delta}$ is generated according to the given probability density function $f_{\mathbf{\Delta}}(\Delta)$. With this generated uncertainty the convex function $U(x, \Delta^{(i)})$ is parametrized and used at the second step of the algorithm where 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. Using some existing facts, and 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. To initialize the algorithm, a method is presented for obtaining an initial ellipsoid that contains the solution set. 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.

3 The Ellipsoid Algorithm (EA)

The algorithm presented below releases the restrictive Assumption 1, and retains only Assumption 2. Convergence in a finite number of iterations with probability one is also guaranteed.

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

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

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 [2,7].

Suppose that after iteration i 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 Ellipsoid algorithm is then summarized as follows.

Algorithm 2 (The Ellipsoid Algorithm: iteration $i+1$) *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.*

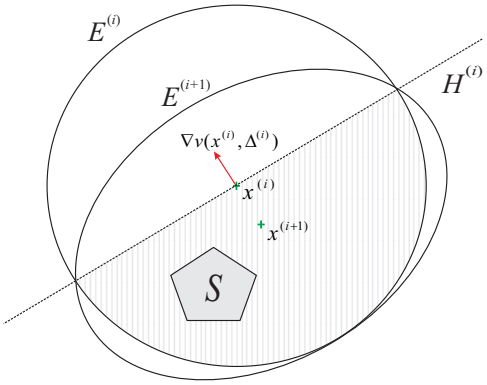

Fig. 2. One iteration of the ellipsoid method in the two-dimensional case.

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

$$x^{(i+1)} = \begin{cases} x^{(i)} - \frac{1}{N+1} \frac{P_i \nabla v(x^{(i)}, \Delta^{(i)})}{\sqrt{\nabla^T v(x^{(i)}, \Delta^{(i)}) P_i \nabla v(x^{(i)}, \Delta^{(i)})}} & \text{if } v(x^{(i)}, \Delta^{(i)}) \neq 0, \\ x^{(i)} & \text{if } v(x^{(i)}, \Delta^{(i)}) = 0, \end{cases}$$

$$P_{i+1} = \begin{cases} \frac{N^2}{N^2-1} \left(P_i - \frac{2}{N+1} \frac{P_i \nabla v(x^{(i)}, \Delta^{(i)}) \nabla^T v(x^{(i)}, \Delta^{(i)}) P_i^T}{\nabla^T v(x^{(i)}, \Delta^{(i)}) P_i \nabla v(x^{(i)}, \Delta^{(i)})} \right) & \text{if } v(x^{(i)}, \Delta^{(i)}) \neq 0, \\ P_i & \text{if } v(x^{(i)}, \Delta^{(i)}) = 0, \end{cases} \tag{11}$$

The initialization of the EA as well as the stopping criterion remain the same as for the SIA.

Figure 2 visualizes Algorithm EA in the two-dimensional case. The convergence of the approach is established immediately, provided that Assumption 2 holds, which implies that for any $x^{(i)} \notin \mathcal{S}$ there exists a non-zero probability for the execution of a correction step (i.e. there is a non-zero probability for generation of $\Delta^{(i)} \in \mathbf{\Delta}$ such that $v(x^{(i)}, \Delta^{(i)}) > 0$).

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

(i) $\mathbf{vol}(\mathcal{S}) > 0$. *Then a feasible solution will be found in a finite number of iterations with probability one.*

(ii) $\mathbf{vol}(\mathcal{S}) = 0$. *Then*

$$\lim_{i \to \infty} x^{(i)} = x^* \in \mathcal{S}$$

with probability one.

PROOF. Algorithm EA generates ellipsoids with geometrically decreasing volumes [2], i.e. for the i-th correction step we can write

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

Due to Assumption 2, for any $x^{(i)} \notin \mathcal{S}$ there exists a non-zero probability for the execution of a correction step (i.e. there is a non-zero probability, independent on the iteration number i, for generation of $\Delta^{(i)} \in \mathbf{\Delta}$ such that $v(x^{(i)}, \Delta^{(i)}) > 0$). Therefore

$$\lim_{i \to \infty} \mathbf{vol}(E^{(i)}) = 0. \tag{12}$$

(i) If we then suppose that the solution set $\mathcal{S}$ has a non-empty interior, i.e. $\mathbf{vol}(\mathcal{S}) > 0$, then from Equation (12) and due to the fact that $E^{(i)} \supseteq \mathcal{S}$ for all $i = 0, 1, \ldots$, 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 $\mathbf{vol}(\mathcal{S}) = 0$, then due to the convexity of the function, and due to Equation (12), the algorithm will converge to a point in $\mathcal{S}$ with probability one. □

The result in Lemma 2 outlines the advantages of Algorithm EA over the previously proposed Algorithm SIA. While in the case $\mathbf{vol}(\mathcal{S}) > 0$ Algorithm EA preserves the property of guaranteed convergence with probability one in a finite number of iterations, it offers the advantages over Algorithm SIA that

+ no a-priori knowledge about a number $r > 0$ satisfying the condition in Assumption 1 is necessary (we will discuss how to find an initial ellipsoid in the next Section), and
+ it converges (although at infinity) even in the case that the set $\mathcal{S}$ has an empty interior.

Finally, similarly to the bound I_{SIA} on the maximum number of correction steps for the Subgradient Iteration Algorithm (see Equation (7)), we can derive such an upper bound for the proposed Ellipsoid method.

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

$$I_{EA} = 2N \left\lceil \ln \frac{\mathbf{vol}(E^{(0)})}{\mathbf{vol}(\mathcal{S})} \right\rceil \tag{13}$$

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 $\lceil a \rceil$, $a \in \mathbb{R}$, denotes the minimum integer number larger than or equal to a.

PROOF. It is shown in [2] that for the i-th correction step one can write

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

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

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

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

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

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

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

or

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

Therefore, Equation (13) is proven. □

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

Example 4 (Comparison between the bounds 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 $x^* \in \mathbb{R}^{10}$*

$$\mathcal{S} = \{x \in \mathbb{R}^{10} : \ \|x - x^*\| \leq 1.1\}.$$

To make a fair comparison between the SIA and the newly proposed EA we proceed as follows: we assume that the initial condition $x^{(0)}$ for SIA is at a distance

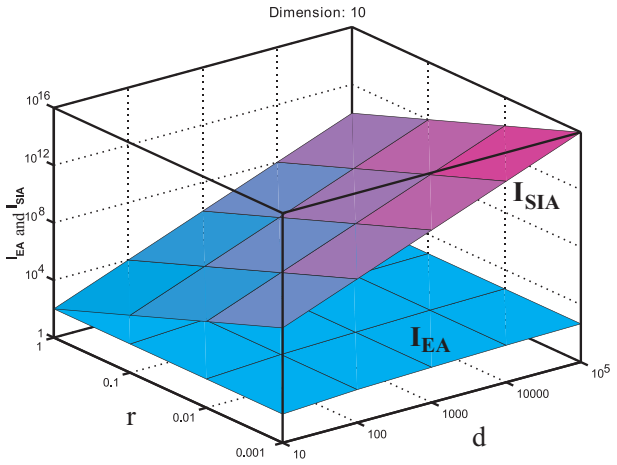

Fig. 3. Comparison between the upper bounds I_{EA} and I_{SIA} for the algorithms SIA and EA.

$d > 1.1$ from the center of $\mathcal{S}$, i.e. $\|x^{(0)} - x^*\| = d$, and that the initial ellipsoid for 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 3 represents the results (note that all the three axes are in logarithmic scale). Clearly, $I_{EA} \ll I_{SIA}$. It should be pointed out that even if one selects the initial ellipsoid for the EA to be a ball of radius $10d$, or even $100d$, one still gets $I_{EA} \ll I_{SIA}$.

In the next Section we present a method to obtain an initial ellipsoid.

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

Before the method for obtaining an initial ellipsoid is presented, some additional notation must be introduced. In addition to the solution set $\mathcal{S}$ and the level sets $LS(c, \Delta)$, we now define the *local solution sets* for any fixed $\Delta_i \in \mathbf{\Delta}$ as the level set at zero

$$S_{\Delta_i} \doteq LS(0, \Delta_i). \tag{14}$$

Therefore, any $x^* \in \mathcal{S}$ is such that $x^* \in S_\Delta$ for all $\Delta \in \mathbf{\Delta}$. Also the solution set $\mathcal{S}$ is the intersection of all local solution sets

$$\mathcal{S} = \bigcap_{\Delta_i \in \mathbf{\Delta}} S_{\Delta_i}.$$

Note also, that $LS(c, \Delta) \supseteq S_\Delta \supseteq \mathcal{S}$. Due to the convexity of the functions

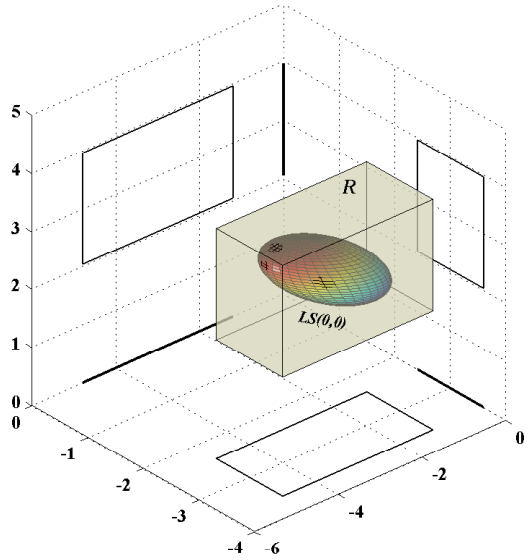

Fig. 4. The initial ellipsoid is computed by first bounding the level set $LS(0,0)$ with a box, and then obtaining an ellipsoid that embraces it (not drawn on the figure).

$v(x, \Delta_i)$ (consult Lemmas 4 and 5 in [3]), the solution set is clearly convex.

The following additional assumption needs to be imposed.

Assumption 3 *It is assumed that the level set $LS(0,0)$ is a bounded set.*

It must be noted that this assumption is not very restrictive since we are anyway not interested in unbounded solutions because the solutions are later on needed for controller/observer parametrization. For instance, the optimal state-feedback gain F in Example 1 is parametrized by any solution $\{X, Y\}$ to the LMI in Equation (9) as $F = YX^{-1}$, and thus unbounded solutions are clearly of no interest. Whenever this assumption does not hold, it can be enforced by introducing additional hard constraints on the entries of x. Such constraints can be directly included into the algorithm for initial ellipsoid computation that is presented below.

The goal is to find an ellipsoid containing the solution set $\mathcal{S}$. For this purpose we will make use of the fact that $\mathcal{S}$ is contained in any local solution set S_Δ, and therefore in any level set $LS(c, \Delta)$ for any $c > 0$ and $\Delta \in \mathbf{\Delta}$. It is, therefore, contained in $LS(0,0)$, i.e. $\mathcal{S} \subseteq LS(0,0)$. The idea is then to find an ellipsoid that contains the level set $LS(0,0)$. To this end we will first bound the set $LS(0,0)$ with a rectangular parallelepiped, and then we build an ellipsoid around it as shown in Figure 4, which we will use as an initial ellipsoid to start Algorithm EA. In order to find a bounding rectangular parallelepiped,

we need to find solutions to the following constrained optimization problems

$$\bar{x}_i = \max_{x \in \mathcal{X}} x_i, \text{ subject to } x \in LS(0,0), \ i = 1, 2, \ldots, N,$$
$$\underline{x}_i = \min_{x \in \mathcal{X}} x_i, \text{ subject to } x \in LS(0,0), \ i = 1, 2, \ldots, N,$$

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*

$$\bar{x}_i = \max_{x \in \mathcal{X}} x_i, \ \textit{subject to } U(x,0) \leq 0, \ i = 1, 2, \ldots, N,$$
$$\underline{x}_i = \min_{x \in \mathcal{X}} x_i, \ \textit{subject to } U(x,0) \leq 0, \ i = 1, 2, \ldots, N.$$

(b) *Take* $\bar{x} = [\bar{x}_1, \ldots, \bar{x}_N]^T$ *and* $\underline{x} = [\underline{x}_1, \ldots, \underline{x}_N]^T$*, 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* $P = \mathbf{diag}\frac{1}{2}(\bar{x} - \underline{x})^2$ *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

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

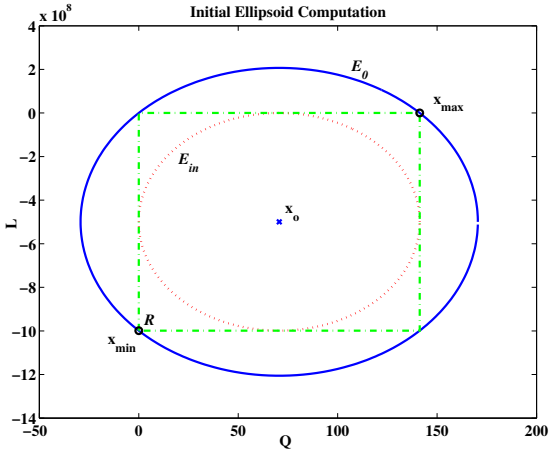

Fig. 5. Illustration of the algorithm for initial ellipsoid computation.

Example 5 (Initial Ellipsoid Computation) *To illustrate the algorithm for initial ellipsoid computation, proposed in the previous Section, we consider the following system*

$$\dot{x}(t) = -x(t) + u(t) + \xi(t)$$
$$z(t) = x(t)$$

for which a constant state-feedback controller has to be designed such that the $\mathcal{H}_\infty$-norm of the resulting closed-loop system is less than $\gamma = 10^{-5}$. Using the results in [2], this would be the case if there exist $Q \in \mathbb{R}$, $R \in \mathbb{R}$, and $L \in \mathbb{R}$ such that

$$\begin{bmatrix} Q & 0 & 0 & 0 \\ 0 & 2Q - L - L^T & 1 & Q \\ 0 & 1 & 1 & 0 \\ 0 & Q & 0 & \gamma \end{bmatrix} > 0$$

Figure 5 visualizes the initial ellipsoid that was generated by Algorithm 3.

5 Experimental part

Next, we present an example illustrating the probabilistic approach developed in this paper used to design a robust $\mathcal{H}_2$ state-feedback controller for a model, representing a real-life diesel actuator benchmark system, taken from [1]. A

linear, continuous-time model of the system can be written in state-space form as

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} 0 & -\frac{K_v}{T_v} & 0 \\ \frac{K_q\eta}{I_{tot}} & -\frac{f_{tot}+K_vK_q\eta}{I_{tot}} & 0 \\ 0 & \frac{1}{N} & 0 \end{bmatrix} x(t) + \begin{bmatrix} \frac{K_v}{T_v} \\ \frac{K_vK_q\eta}{I_{tot}} \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ \frac{1}{NI_{tot}} \\ 0 \end{bmatrix} \xi(t) \\ z(t) &= [0,\ 1,\ 0]x(t)\end{aligned} \tag{15}$$

The values of the parameters in the state-space model above are $\eta \in [\,0.7,\ 0.85\,]$, $f_{tot} \in [\,9.85 \times 10^{-3},\ 5.91 \times 10^{-2}\,]$, $I_{tot} \in [\,2.1505 \times 10^{-3},\ 2.9095 \times 10^{-3}\,]$, $K_q \in [\,0.513,\ 0.567\,]$, $K_v = 0.9$, $N = 89$, $\alpha_s = 0.987$, and $T_v = 8.8 \times 10^{-3}$. Note, that four of the parameters are uncertain.

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 [2]: Find matrices $Q = Q^T$, $R = R^T$, and L such that for all possible values of the parameters θ

$$\begin{aligned} \operatorname{trace}(R) &< 1 \\ \begin{bmatrix} R & C_zQ \\ QC_z^T & Q \end{bmatrix} &> 0 \\ \begin{bmatrix} -A(\theta)Q - QA(\theta)^T - B_u(\theta)L - L^TB_u(\theta)^T & B_\xi(\theta) \\ B_\xi(\theta)^T & I \end{bmatrix} &> 0 \end{aligned}$$

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 approach was proposed to the probabilistic design of robust controllers (state estimators), based on the Ellipsoid Algorithm. It features a number of advantages over the probabilistic Subgradient Iteration Algorithm, recently proposed in [12,3]. Although the latter possessed a number of useful properties, namely guaranteed convergence in a finite number of iterations with probability one, applicability to general uncertainty structures and to large numbers of uncertain parameters, it has the strong disadvantage that the radius of a non-empty ball contained in the solution set must be known. This drawback is removed in the EA approach proposed in this paper, while still retaining the advantages of the SIA method. Similarly to the SIA method, at each iteration of the EA two steps are performed. In the first step a random uncertainty sample $\Delta^{(i)} \in \mathbf{\Delta}$ is generated according to the given probability density function $f_{\mathbf{\Delta}}(\Delta)$. With this generated uncertainty a suitably defined convex function is parametrized so that at the second step of the algorithm an ellipsoid is computed, in which the solution set is guaranteed to lie. As a result, the EA algorithm produces a sequence of ellipsoids with decreasing volumes, all containing the solution set. An efficient method for obtaining an initial ellipsoid is also proposed in the paper. The approach is illustrated by means of a case study with a real-life diesel actuator benchmark model with four real uncertain parameters, for which an $\mathcal{H}_2$ robust state-feedback controller was designed.

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 Real and Complex Structured Uncertainty, *IEEE Transactions on Automatic Control*, 45(12), 2000, pp. 2218-2235.

[5] X. Chen, K. Zhou, Order Statistics and Probabilistic Robust Control, *Systems & Control Letters*, 35(3), 1998, pp. 175-182.

[6] Y. Fujisaki, F. Dabbene, R. Tempo, Probabilistic Robust Design of LPV Control Systems, *Procedings of the 40th IEEE Conference on Decision and Control CDC 2001*, Orlando, Florida, pp. 2019-2024.

[7] M. Grötschel, L. Lovász, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization,* Springer-Verlag, Berlin, Germany, 1988.

[8] 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.

[9] M. Kothare, V. Balakrishnan, M. Morari, Robust Constrained Model Predictive Control using Linear Matrix Inequalities, *Automatica*, 32(10), 1996, pp. 1361-1379.

[10] 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.

[11] H. Kushner, J. Yin, Stochastic Approximation Algorithms and Applications, Springer-Verlag, New York, 1997.

[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] R. Tempo, F. Dabbene, Randomized Algorithms for Analysis and Control of Uncertain Systems: An Overview, *in Perspectives in Robust Control - Lecture Notes in Control and Information Science ,(ed. S.O. Moheimani)*, Springer-Verlag, London, pp. 347-362, 2001.

[15] V. Ugrinovskii, Randomized Algorithms for Robust Stability and Control of Stochastic Hybrid Systems with Uncertain Switching Policies, *Proceedings of the 40th IEEE Conference on Decision and Control CDC 2001*, Orlando, Florida, pp. 5026-5031.

[16] M. Vidyasagar, Statistical Learning Theory and Randomized Algorithms for Control, *IEEE Control Systems Magazine*, 18(6), 1998, pp. 69-85.