Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

$\mathcal{H}_\infty$ Guaranteed Cost Computation for Uncertain Time-Delay Systems

E. N. Gonçalves, S. B. Bastos, R. M. Palhares, R. H. C. Takahashi, R. C. Mesquita, C. D. Campos and P. Ekel

Abstract— **This paper presents a less conservative approach to compute, with any prescribed accuracy, the $\mathcal{H}_\infty$ guaranteed cost of time-delay continuous-time linear time-invariant systems subjected to polytopic uncertainties. The proposed analysis approach is based on a branch-and-bound algorithm that incorporates a recent LMI-based analysis formulation and a new polytope partition strategy. In the branch-and-bound algorithm, the upper bound function is defined as the worst case guaranteed $\mathcal{H}_\infty$ disturbance attenuation level computed for the subpolytopes achieved with successive partitions of the polytope which describes the uncertainty domain. The lower bound function is defined as the worst case $\mathcal{H}_\infty$ norm computed in the polytope and subpolytope vertices. The difference between the upper and lower bound functions converges to zero as the initial polytope is split into smaller subpolytopes resulting in the $\mathcal{H}_\infty$ guaranteed cost for the whole initial polytope with the required accuracy. It is also presented an algorithm to implement a $d-$dimensional simplex subdivision technique to be used in the branch-and-bound algorithm.**

Index Terms— **$\mathcal{H}_\infty$ cost computation, time-delay, polytope-bounded uncertainty, branch-and-bound algorithm, simplex edgewise subdivision**

I. Introduction and Motivation

The stability and performance analysis problem of linear time-invariant systems subjected to parametric uncertainties and time-delay had received a lot of attention in the control community, reaching several areas, from the academic area to the industrial area as well. Many approaches have been proposed to deal with this problem and are available in the literature. The recent most effective approaches have been formulated as an optimization or a feasibility problem described in terms of linear matrix inequality (LMI) [1]–[6].

Even though stability and performance analysis approaches based on LMI are proven to be very flexible, they are still limited. Therefore, conservative results may be generated.

This work has been supported in part by CNPq and FAPEMIG, Brazil

E. N. Gonçalves is with Department of Electrical Engineering, Federal Center of Technological Education of Minas Gerais, Av. Amazonas 7675, 31510-470, Belo Horizonte, MG, Brazil eduardong@des.cefetmg.br

S. B. Bastos and R. M. Palhares are with Department of Electronics Engineering, Federal University of Minas Gerais, Av. Antônio Carlos 6627, 31270-010, Belo Horizonte, MG, Brazil palhares@cpdee.ufmg.br

R. H. C. Takahashi is with Department of Mathematics, Federal University of Minas Gerais, Av. Antônio Carlos 6627 - 31270-010, Belo Horizonte, MG, Brazil

R. C. Mesquita is with Department of Electrical Engineering, Federal University of Minas Gerais, Av. Antônio Carlos 6627, 31270-010, Belo Horizonte, MG, Brazil

C. D. Campos and P. Ya. Ekel are with Graduate Program in Electrical Engineering, Pontifical Catholic University of Minas Gerais, Av. Dom José Gaspar 500, 30535-610, Belo Horizonte, MG, Brazil

The main contribution of this work is to introduce a new less conservative approach to compute the $\mathcal{H}_\infty$ guaranteed cost which could be used as a performance/stability analysis tool for dynamic systems subjected to uncertainties and time-delay according to the following representation:

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + A_d x(t-\tau) + Ew(t) \\ z(t) & = & Cx(t) + Fw(t) \end{array} \right. \tag{1}$$

where matrices A, A_d, E, C, and F belong to the polytope:

$$\begin{array}{rl} \mathcal{T}(\alpha) = & \Big\{ (A, A_d, E, C, F) = \\ & \displaystyle\sum_{i=1}^{N} \alpha_i \left(A_i, \ldots, F_i \right), \alpha \in \Omega \Big\} \end{array} \tag{2}$$

with $\Omega \triangleq \left\{ \alpha \ : \ \alpha_i \geq 0 \ , \ \sum_{i=1}^{N} \alpha_i = 1 \right\}$, N the number of polytope vertices, and $\alpha = [\ \alpha_1 \ \ldots \ \alpha_N]^T$ the vector that parametrizes the polytope.

The representation by polytopic uncertainties can be obtained from systems with different models for each operation point, from non-linear systems, and also from systems with not precisely known parameters.

The proposed approach to compute less conservative $\mathcal{H}_\infty$ guaranteed cost for uncertain time-delay linear time-invariant systems is based on the $\mathcal{H}_\infty$ performance analysis presented in [1] and on the branch-and-bound algorithm. The branch-and-bound algorithm has been already applied in the robust control area, e.g. as a possibility to reduce conservatism of robust stability analysis of linear systems with real uncertain time-invariant parameters [7], to compute the minimum norm of a linear fractional transformation [8], to solve biaffine matrix inequality (BMI) feasibility problems [9], and to solve $\mathcal{H}_2$-norm model reduction problems [10]. In section II, it is presented a general description of the branch-and-bound algorithm. In sections III and IV, it is presented a detailed implementation of the proposed branch-and-bound algorithm where the subdivision operation will be based on the Delaunay triangulation followed by an edgewise subdivision of simplex.

II. The Branch-and-Bound Algorithm

The following description of the branch-and-bound (BnB) algorithm is adapted from [7]. The BnB algorithm can be used to compute the global maximum of a function $f(p)$: $\mathbb{R}^d \to \mathbb{R}$, with a prescribed accuracy $\epsilon > 0$, where the domain is defined as a d-dimensional hyper-rectangle

$$\mathcal{P}_{init} = [\underline{p}_1, \overline{p}_1] \times [\underline{p}_2, \overline{p}_2] \times \ldots \times [\underline{p}_d, \overline{p}_d]$$

where $\underline{p}_i$ and $\overline{p}_i$, $i = 1, \ldots, d$, are the extreme values of the entries of the parameter vector $p = [p_1\ p_2\ \ldots\ p_d]^T$, i.e., $p_i \in [\underline{p}_i, \overline{p}_i]$.

For an hyper-rectangle $\mathcal{P} \subseteq \mathcal{P}_{init}$, one can define

$$\Phi_{max}(\mathcal{P}) \triangleq \max_{p \in \mathcal{P}} f(p) \tag{3}$$

The BnB algorithm calculates $\Phi_{max}(\mathcal{P}_{init})$ based on two functions, $\Phi_{lb}(\mathcal{P})$ and $\Phi_{ub}(\mathcal{P})$, defined over $\{\mathcal{P} : \mathcal{P} \subseteq \mathcal{P}_{init}\}$. These two functions must satisfy the following conditions:

$$\Phi_{lb}(\mathcal{P}) \leq \Phi_{max}(\mathcal{P}) \leq \Phi_{ub}(\mathcal{P}) \tag{4}$$

$$\begin{array}{c} \forall\, \epsilon > 0,\ \exists\, \delta > 0 \text{ such that } \forall\, \mathcal{P} \subseteq \mathcal{P}_{init}, \\ dim(\mathcal{P}) \leq \delta \Rightarrow \Phi_{ub}(\mathcal{P}) - \Phi_{lb}(\mathcal{P}) \leq \epsilon \end{array} \tag{5}$$

Condition (4) states that $\Phi_{lb}(\mathcal{P})$ and $\Phi_{ub}(\mathcal{P})$ functions calculate the lower and upper bounds of $\Phi_{max}(\mathcal{P})$, respectively. Condition (5) states that, as the maximum edge length of $\mathcal{P}$, denoted by $dim(\mathcal{P})$, tends to zero, the difference between the lower and upper bounds converges to zero.

The BnB algorithm starts from the calculation of $\Phi_{lb}(\mathcal{P}_{init})$ and $\Phi_{ub}(\mathcal{P}_{init})$. If $(\Phi_{ub}(\mathcal{P}_{init}) - \Phi_{lb}(\mathcal{P}_{init}))/\Phi_{lb}(\mathcal{P}_{init}) \leq \varepsilon$, where ε is a required relative accuracy, then the algorithm ends. If the stop criterion is not achieved, it is necessary to subdivide the hyper-rectangle $\mathcal{P}_{init}$ into a set of smaller hyper-rectangles, $\mathcal{P}_{init} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_S$, and calculate $\Phi_{lb}(\mathcal{P}_i)$ and $\Phi_{ub}(\mathcal{P}_i)$, $i = 1, \ldots, S$. Therefore

$$\max_{1 \leq i \leq S} \Phi_{lb}(\mathcal{P}_i) \leq \Phi_{max}(\mathcal{P}_{init}) \leq \max_{1 \leq i \leq S} \Phi_{ub}(\mathcal{P}_i)$$

supplies the new bounds for $\Phi_{max}(\mathcal{P}_{init})$. If the relative difference between the new bounds is smaller or equal to ε, then the algorithm ends, else the partition of $\mathcal{P}_{init}$ is refined again and new bounds are computed. The BnB algorithm converges, since $dim(\mathcal{P}_i)$, $i = 1, \ldots, S$, tends to zero and the hyper-rectangle tends to a point, causing $\Phi_{ub}(\mathcal{P}_{init}) - \Phi_{lb}(\mathcal{P}_{init})$ to tend to zero.

The version of the BnB algorithm used, adapted from [7], is stated as follows:

Algorithm *Branch-and-Bound*
- $k \leftarrow 0$;
- $\mathcal{L}_0 \leftarrow \{\mathcal{P}_{init}\}$;
- $L_0 \leftarrow \Phi_{lb}(\mathcal{P}_{init})$;
- $U_0 \leftarrow \Phi_{ub}(\mathcal{P}_{init})$;
- **while** $(U_k - L_k)/L_k > \varepsilon$
  - select $\mathcal{P} \in \mathcal{L}_k$ such that $\Phi_{ub}(\mathcal{P}) = U_k$;
  - split $\mathcal{P}$ in $\mathcal{P}_1, \ldots, \mathcal{P}_S$
  - $\mathcal{L}_{k+1} \leftarrow \{\mathcal{L}_k - \mathcal{P}\} \cup \{\mathcal{P}_1, \ldots, \mathcal{P}_S\}$;
  - $L_{k+1} \leftarrow \max_{\mathcal{P} \in \mathcal{L}_{k+1}} \Phi_{lb}(\mathcal{P})$;
  - $U_{k+1} \leftarrow \max_{\mathcal{P} \in \mathcal{L}_{k+1}} \Phi_{ub}(\mathcal{P})$;
  - eliminate all $\mathcal{P} \in \mathcal{L}_{k+1}$ such that
    - $\Phi_{ub}(\mathcal{P}) < L_{k+1}$;
  - $k \leftarrow k + 1$;
- **end while**

end algorithm

III. The Less Conservative Approach to Compute the $\mathcal{H}_\infty$ Disturbance Attenuation Level

In order to apply the BnB algorithm into a less conservative approach to compute the $\mathcal{H}_\infty$ guaranteed cost, γ_a, with a required accuracy, it is necessary to find out a lower bound function $\Phi_{lb}(\mathcal{P})$ and a upper bound function $\Phi_{ub}(\mathcal{P})$ that satisfy conditions (4) and (5).

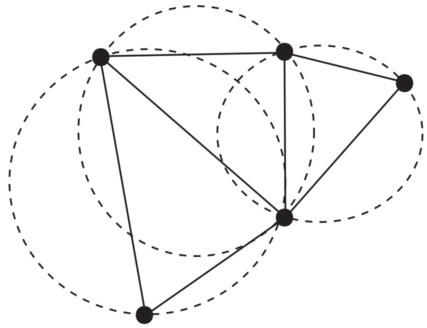
As the guaranteed cost over the polytope is never lower than the norm in a point of the polytope and as the guaranteed cost tends to the value of the norm as the polytope tends to a point (maximum polytope edge tends to zero), the $\mathcal{H}_\infty$ guaranteed cost and the worst case $\mathcal{H}_\infty$ norm are qualified functions to be employed as upper and lower bound functions respectively. The worst case $\mathcal{H}_\infty$ norm is computed in a finite number of polytope points, v, initialized as the polytope vertices, with $N \leq v < \infty$. As the initial polytope is subdivided in smaller polytopes, new vertices are generated and included in the set which is computed the worst case $\mathcal{H}_\infty$ norm. Since only the polytope with the worst case $\mathcal{H}_\infty$ guaranteed cost is subdivided at each iteration, the lower bound function works as a smart grid technique where the refinements are performed only where it is necessary.

In this way, to implement the proposed analysis approach, it is necessary formulations to compute the $\mathcal{H}_\infty$ guaranteed cost of a uncertain time-delay linear time-invariant system, with polytope-bounded uncertainty, and the $\mathcal{H}_\infty$ norm of a precisely known time-delay linear system. Based on the results presented in [1], it is possible to determine the lower and upper functions by means of the the following optimization problem:

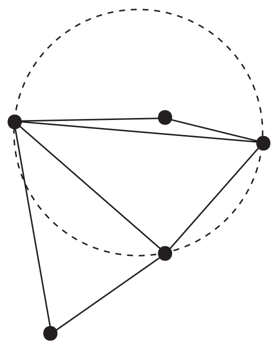
$$\left\{ \begin{array}{l} \gamma = \sqrt{\displaystyle\min_{\lambda, X, H, Q, ZeV} \lambda} \\ \text{subject to:} \\ \begin{bmatrix} \Upsilon_i & XA_{di} - V & XE_i & \bar{\tau} A_i^T Z & C_i^T \\ * & -Q & \mathbf{0} & \bar{\tau} A_{di}^T Z & \mathbf{0} \\ * & * & -\lambda \mathbf{I} & \bar{\tau} E_i^T Z & F_i^T \\ * & * & * & -\bar{\tau} Z & \mathbf{0} \\ * & * & * & * & -I \end{bmatrix} \prec \mathbf{0} \\ \begin{bmatrix} H & V \\ * & Z \end{bmatrix} \succeq \mathbf{0}, \quad \forall i = 1, \ldots, N \\ \Upsilon_i \triangleq XA_i + A_i^T X + \bar{\tau} H + V + V^T + Q \end{array} \right. \tag{6}$$

where $\bar{\tau}$ is the upper bound on state-delay and γ is the $\mathcal{H}_\infty$ norm in the case of a precisely known system ($N = 1$), or the $\mathcal{H}_\infty$ guaranteed cost in the case of a uncertain system ($N > 1$).

The partition of $\mathcal{P}_{init}$ in the BnB algorithm can be implemented by several different techniques. In this paper, the polytope is initially decompose in a set of simplices applying the Delaunay triangulation. The Delaunay triangulation subdivides a 2-dimensional region into triangles (tetrahedra in 3-dimensional or simplices in d-dimensional spaces). The Delaunay triangulation maximizes the minimal angle between edges for all possible triangulations of a set of points. A simplex achieved with the Delaunay triangulation determines a hypersphere that does not contain any other



(a) Delaunay triangulation.



(b) General triangulation.

Fig. 1. Triangulations of 5 points in the 2-d space.

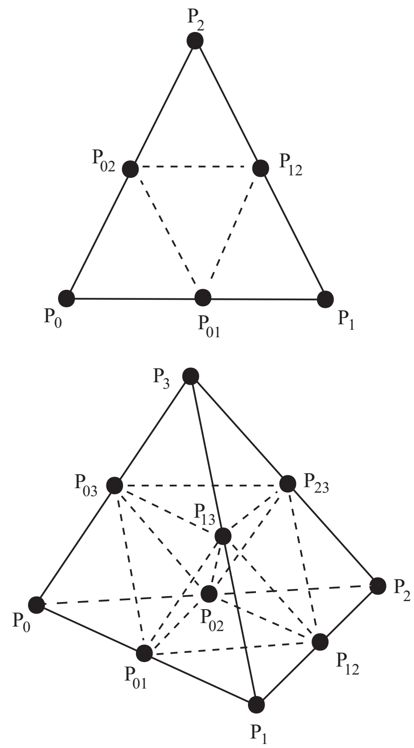


Fig. 2. Triangle and tetrahedron partitions using an edgewise subdivision.

point of the set of points, unless the $d+1$ vertices of the simplex (see Fig. 1). There is a straight relation between the Delaunay triangulation of a set of points and the convex hull of the *lifting transformation* [11] of these points in one higher dimension. Therefore, algorithms to compute the convex hull in (d+1)-dimensional spaces can be used to compute the Delaunay triangulation in d-dimensional spaces. This is done in the MATLAB® function $delaunayn(\cdot)$ that is based on the Quickhull algorithm [12].

After the initial triangulation of $\mathcal{P}_{init}$, succeeding refinements will be accomplished by a simplex edgewise subdivision technique [13]. Subdivisions are obtained by the introduction of new points over the edges of the simplex $\mathcal{P}$ which fulfills the condition $\Phi_{ub}(\mathcal{P}) = U_k$. These new points supply the conditions to subdivide $\mathcal{P}$ into 2^d new simplices. The 2^d edgewise subdivision of a triangle and a tetrahedron are shown in Fig. 2, where $P_{ij} \triangleq (P_i + P_j)/2$. There are many advantages in applying the simplex edgewise subdivision [13]. Two of them have special importance to the considered application. First, all simplices achieved with the subdivision have the same volume, which guarantees that the volume tends to zero with successive subdivisions. Second, the number of congruence classes of simplices that results from successive refinements are limited to $d!/2$, which is the optimum value for simplex subdivision [13], [14]. This feature means that this subdivision technique avoids to create degenerated simplices, i.e., simplices with too small angles between edges. This property operates in favor of the branch-and-bound algorithm convergence. Notice that, differently from other Engineering applications, such as finite elements, there is no concern in guaranteeing the division consistency by not using simplex vertices over the edge of another one. The proposed algorithm to implement a simplex edgewise subdivision will be introduced in the next section.

The advantage provided by the combination of Delaunay triangulation and simplex subdivision is that it allows the proposed analysis approach be employed with polytopes of general shape not limited to hyper-rectangles. Another advantage of dealing with simplices is that a simplex is the polytope with lower number of vertices. This feature is useful to reduce the computational time to compute the $\mathcal{H}_\infty$ guaranteed cost with formulation (6).

IV. Simplex Edgewise Subdivision

The algorithm proposed on this section implements an edgewise subdivision of a d-dimensional simplex into k^d simplices, being inspired by the color scheme concept presented in [13]. Much of the following notation and terminology is from [13]. Consider a d-simplex σ defined as a $d+1$ points sequence, $P_0, P_1, \ldots, P_d$, which are independent in $\mathbb{R}^d$. Consider the notation

$$P_{\chi_1\chi_2\cdots\chi_k} = (P_{\chi_1} + P_{\chi_2} + \ldots + P_{\chi_k})/k \qquad (7)$$

with $\chi_i \in \{0, 1, \ldots, d\}$, $i = 1, \ldots, k$.

The edgewise subdivision of σ into k^d simplices will be obtained from the simplex vertices $P_0, P_1, \ldots, P_d$ and from new points $P_{\chi_1\chi_2\cdots\chi_k}$, as defined in (7). The points that define each new simplex will be obtained from a matrix $M \in \mathbb{N}^{k\times(d+1)}$, denominated color scheme, which entries are integer numbers in the set $\{0, 1, \ldots, d\}$, denominated colors, which represent the indices of the vertices $P_0, P_1, \ldots, P_d$ [13]. The i-th column of M will define the i-th point $P_{\chi_{0,i}\chi_{1,i}\cdots\chi_{k-1,i}}$ of the new simplex. In order to improve the readability of the algorithm that will be proposed, the lines of M are numbered from 0 instead of from 1 as in [13]. The main characteristics of the color scheme are that its elements appear in a non-decreasing order when they are read as text:

$$\chi_{0,0} \leq \chi_{0,1} \leq \ldots \leq \chi_{0,d} \leq \chi_{1,0} \leq \ldots \leq \chi_{k-1,d}$$

and the *i*-th column differs from the (*i*-1)th column by an unitary increment in only one entry.

The problem treated here is how to obtain the k^d color schemes to generate the complete simplex subdivision. In this section, it is proposed a read-to-implement algorithm to accomplish the task of automatically generate all the color schemes. Consider $\chi^{\mathrm{n}}_{\mathrm{i,j}}$ the entry of the i-th row and j-th column of the n-th color scheme, $n = 0, \ldots, k^{\mathrm{d}} - 1$. In the proposed algorithm, the n-th color scheme, M^n, will be created line by line starting from $\chi^n_{0,0} = 0$. In order to know if the next element of the matrix will be kept or will be incremented by one, it is necessary to represent the index n of M^n in a k based numeric system:

$$n = x_{d-1} \times k^{d-1} + x_{d-2} \times k^{d-2} + \ldots + x_0 \times k^0 \qquad (8)$$

The values of the digits x_{d-i}, $i = 1, 2, \ldots, d$, will determine which entry of the (*i*-1)th column will be incremented by one to generate the i-th column. When finishing a row, the next row starts with the last color of the previous row, that is, $\chi^n_{i,0} = \chi^n_{i-1,d}$. The described procedure is implemented by the following proposed algorithm:

Algorithm *Color Scheme*
  for $n = 0, 1, \ldots, k^d - 1$ **do**
    $x_{d-1} \ldots x_0 \leftarrow$ conversion of n to base k;
    $color \leftarrow 0$;
    for $i = 0, 1, \ldots, k-1$ **do**
      $\chi^n_{i,0} \leftarrow color$;
      for $j = 1, \ldots, d$ **do**
        if $x_{d-j} = i$ **then** $color \leftarrow color + 1$;
        $\chi^n_{i,j} \leftarrow color$;
      end
    end
  end
end

Consider, for example, a tetrahedron subdivision in $k^d = 2^3$ sub-tetrahedrons. The color schemes M^n, $n = 0, \ldots, 7$, have k rows and $d+1$ columns:

$$M^n = \begin{bmatrix} \chi^n_{0,0} & \chi^n_{0,1} & \chi^n_{0,2} & \chi^n_{0,3} \\ \chi^n_{1,0} & \chi^n_{1,1} & \chi^n_{1,2} & \chi^n_{1,3} \end{bmatrix}$$

To calculate the sub-tetrahedron with $n = 5$, M^5, the color change will be specified by writing 5 in base 2, i.e., $x = 101_2$, which means that the first color change occur in row 1, the second one in row 0, and the last one in row 1:

$$M^5 = \begin{bmatrix} 0 & 0 & \rightsquigarrow 1 & 1 \\ 1 & \rightsquigarrow 2 & 2 & \rightsquigarrow 3 \end{bmatrix}$$

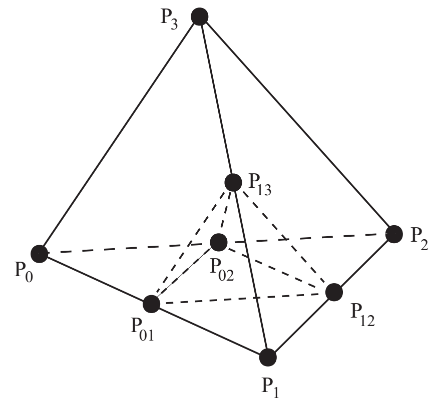


Fig. 3. Sub-tetrahedron generated by a tetrahedron partition in 2^3 parts.

This color scheme shows that the sub-tetrahedron is defined by the set of points $\{P_{01}, P_{02}, P_{12}, P_{13}\}$ as shown in Fig. 3, where point $P_{\chi_{0,i}\chi_{1,i}}$ is calculated by (7).

The abacus presented in [13, Fig. 3], dealing with a 3^7 subdivision, can be generated by the proposed algorithm considering that $1371_{10} = 1 \times 3^6 + 2 \times 3^5 + 1 \times 3^4 + 2 \times 3^3 + 2 \times 3^2 + 1 \times 3^1 + 0 \times 3^0 \equiv 1212210_3$:

$$M^{1371} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 1 & \mathbf{2} & 2 & \mathbf{3} & 3 & 3 & \mathbf{4} & 4 \\ 4 & 4 & \mathbf{5} & 5 & \mathbf{6} & \mathbf{7} & 7 & 7 \end{bmatrix}$$

V. ILLUSTRATIVE EXAMPLE

Consider the uncertain continuous-time linear time-invariant system subjected to a constant time-delay in the state vector:

$$\begin{array}{rcl} \dot{x}(t) & = & Ax(t) + A_d x(t-\tau) + Bu(t) + Ew(t) \\ z(t) & = & Cx(t) \end{array} \qquad (9)$$

with matrices

$$A = \begin{bmatrix} \alpha & 0 \\ 0 & 1+\alpha \end{bmatrix}, \quad A_d = \begin{bmatrix} -1+\beta & -1 \\ 0 & -0.9+\beta \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad E = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$|\alpha| \leq 0.2, \qquad |\beta| \leq 0.2$$

This system has been widely referenced in the literature [1]–[5], [15].

Considering the problem of obtaining a state feedback controller, K, that stabilizes and guarantees the lowest $\mathcal{H}_\infty$ disturbance attenuation level, for the condition of maximum admissible time-delay size, one can obtain, through synthesis theorems proposed in [1], the controller

$$K = \begin{bmatrix} -17.626 & -58.668 \end{bmatrix}$$

which guarantees quadratic stability for the system (9), subjected to time-delays not greater than $\bar{\tau} = 1.1s$ and minimum $\mathcal{H}_\infty$ disturbance attenuation level of $\gamma = 21.03$. Using this controller on the closed loop system, one can assure, with the application of the $\mathcal{H}_\infty$ performance analysis technique

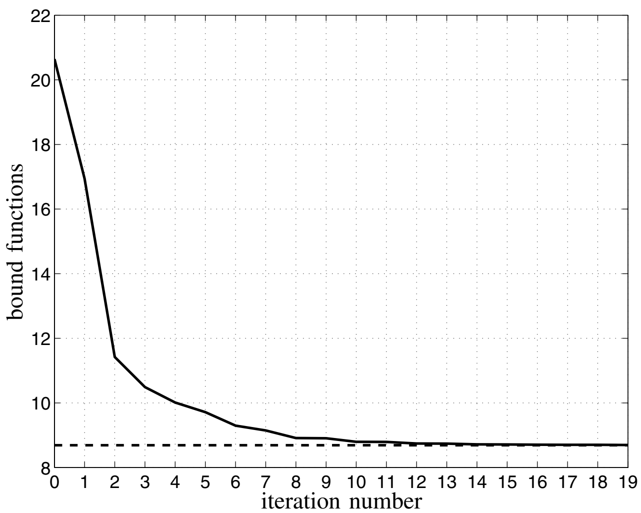


Fig. 4. Evolution of $\Phi_{lb}(\mathcal{P})$ and $\Phi_{ub}(\mathcal{P})$ on the exact calculus of the assured $\mathcal{H}_\infty$ disturbance attenuation level.

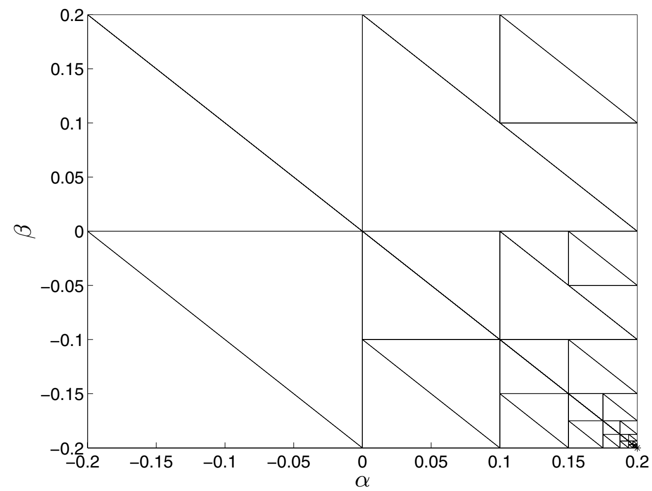


Fig. 5. Partition of $\mathcal{P}_{init}$ after 19 iterations.

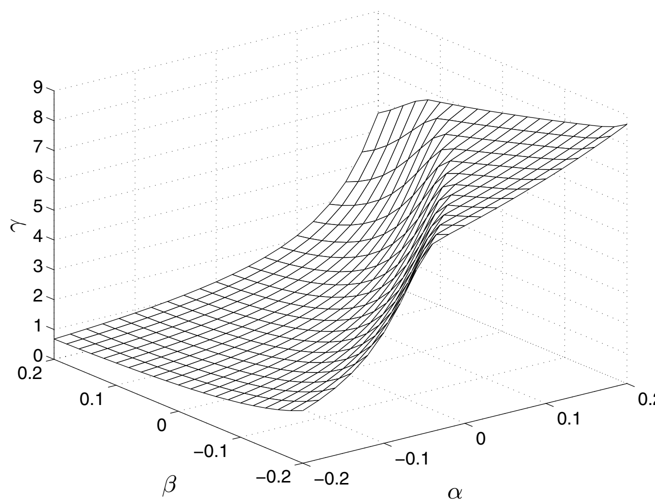


Fig. 6. Values of the level γ of $\mathcal{H}_\infty$ disturbance attenuation, calculated for α and β varying within their extreme values.

described in the works [1] or [2], the $\mathcal{H}_\infty$ guaranteed cost level of $\gamma = 20.42$.

Although the analysis and synthesis techniques, introduced in [1] and [2], allow to attain good performance results for this kind of problem, a certain degree of conservatism still exists on the solutions.

Applying the BnB algorithm to compute the actual $\mathcal{H}_\infty$ guaranteed cost disturbance attenuation level, γ_a, considering the stop criterion $\varepsilon = 0.001$, it is possible to obtain the convergence of the lower bound function $\Phi_{lb}(\mathcal{P})$ and upper bound function $\Phi_{ub}(\mathcal{P})$ to 8.6900 and 8.6965, respectively, after 19 iterations and 32.781s of processing time (Pentium 4 2.8GHz, 512MB computer). Therefore, it can be concluded that $\gamma_a = 8.6965$ is the actual $\mathcal{H}_\infty$ guaranteed cost valid for the whole uncertainty domain of system (9).

Fig. 4 shows the lower and upper bound functions evolution where one can observe how the $\mathcal{H}_\infty$ guaranteed cost computation is gradually improved. Fig. 5 shows the partition of the uncertain space, after 19 iterations. In this case, it is more efficient to parametrize the uncertain domain by the uncertain parameters. The first partition, performed by the $delaunayn(\cdot)$ function, split the rectangle in two triangles. After that, the triangle with worst case guaranteed cost is subdivided in four triangles based on the edgewise simplex subdivision technique. Fig. 6 shows the surface of the $\mathcal{H}_\infty$ disturbance attenuation level computed for the uncertain parameters varying between its limits. One can notice that the actual $\mathcal{H}_\infty$ cost never exceeds the achieved maximum value $\gamma_a = 8.6965$.

To illustrate the usefulness and flexibility of the proposed analysis approach, it will be considered different uncertain specifications in spite of the fact that the controller was not designed considering then. First, instead of polytope-bounded uncertainty, consider norm-bounded uncertainty: $\alpha^2+\beta^2 \leq 1$. To apply the proposed approach, it is considered the norm- to polytope-bounded approximation presented in [16]. In [16], the 2-dimensional norm-bound space (disk) is approximated by the polytope defined by the following N vertices:

$$\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} cos(2\pi i/N) \\ sin(2\pi i/N) \end{bmatrix}, \quad i = 1, \ldots, N. \tag{10}$$

Fig. 7 shows the partition of $\mathcal{P}_{init}$ after 95 iterations and 150.969s of processing time, in the case of norm-bounded uncertainty approximation with $N = 8$ in eq. (10). It is achieved the guaranteed cost of 7.4665. This example proves the capability of the proposed approach to be applied for polytope of any shape.

Finally, consider that the uncertain parameters are in the ranges: $|\alpha| \leq 0.25$, $|\beta| \leq 0.25$. In this case, the $\mathcal{H}_\infty$ performance analysis technique leads to an infeasible problem being incapable to compute the guaranteed cost as expect, since the control gain had not been designed to this larger uncertain parameters. The proposed analysis approach results

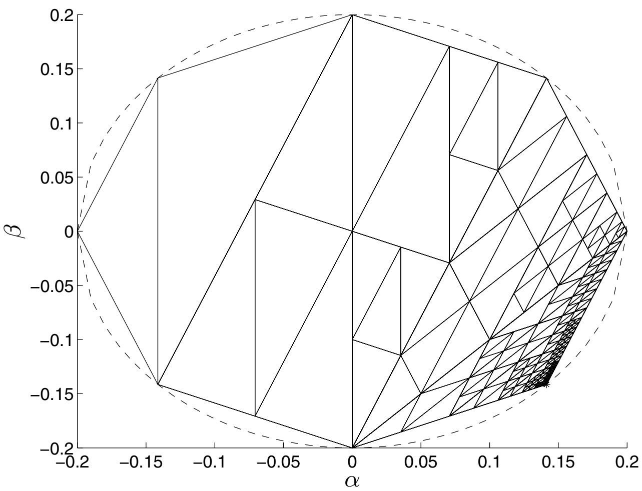


Fig. 7. Partition of $\mathcal{P}_{init}$ after 95 iterations in the case of norm-bounded uncertainty.

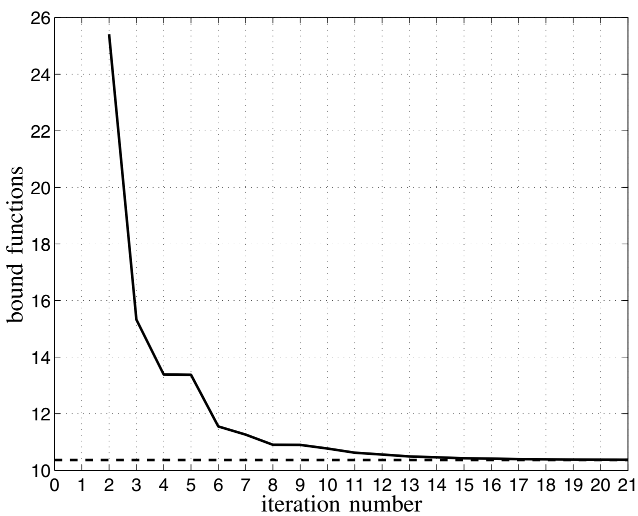


Fig. 8. Evolution of $\Phi_{lb}(\mathcal{P})$ and $\Phi_{ub}(\mathcal{P})$ on the exact calculus of the assured $\mathcal{H}_\infty$ disturbance attenuation level.

in the convergence of the lower bound function $\Phi_{lb}(\mathcal{P})$ and upper bound function $\Phi_{ub}(\mathcal{P})$ to 10.3644 and 10.3721, respectively, for $\varepsilon = 0.001$, after 21 iterations and 35.172s of processing time (see Fig. 8). Only after two partitions of the uncertain space, the LMI-based analysis formulation becomes feasible.

VI. Conclusion

In this paper, it is proposed a new approach to compute the $\mathcal{H}_\infty$ disturbance attenuation level of continuous-time linear time-invariant systems subjected to polytopic uncertainties and time-delay. The proposed approach is based on a combination of the branch-and-bound algorithm and an up-to-date LMI-based formulation that allows to compute the $\mathcal{H}_\infty$ guaranteed cost with a prescribed accuracy. When dealing with systems which have only few uncertain parameters, the algorithm efficiently converges to the value of the cost with the required accuracy. For a greater number of uncertain parameters and/or greater sensitivity of the model to the variation of parameters, it becomes necessary to consider the computational effort involved. Even in these cases in which convergence is slower, the algorithm proposed still should be considered since it provides accurate results under situations in which LMI-based formulations alone are not feasible and when they are, they can produce results too conservative. Besides, it is presented an efficient and simple algorithm to subdivide a $d-$simplex in k^d simplices that can be used, together with the Delaunay triangulation, to implement the partition operation in the branch-and-bound algorithm, which is frequently used in the robust control area. The methodology proposed can be easily extended to the discrete-time case.

References

[1] R. M. Palhares, C. D. Campos, M. C. R. Leles, P. Y. Ekel, and M. F. S. V. D'Angelo, "Delay-dependent robust $\mathcal{H}_\infty$ control of uncertain continuous- and discrete-time linear systems with lumped delays," *IEE Proceedings Control Theory & Applications*, vol. 152, no. 1, pp. 27–34, January 2005.

[2] ——, "Delay-dependent $\mathcal{H}_\infty$ control stabilization of linear systems with time-varying delays," in *Proceedings of the 4th IFAC Workshop on Time-Delay Systems*, Rocquencourt, France, 2003.

[3] E. Fridman and U. Shaked, "A descriptor system approach to $\mathcal{H}_\infty$ control of linear time-delay systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 2, pp. 253–270, 2002.

[4] ——, "Delay-dependent stability and $\mathcal{H}_\infty$ control: constant and time-varying delays," *International Journal of Control*, vol. 76, no. 1, pp. 48–60, 2003.

[5] H. Gao and C. Wang, "Comments and further results on "A descriptor system approach to $\mathcal{H}_\infty$ control of linear time-delay systems"," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 520–525, 2003.

[6] Y. S. Lee, Y. S. Moon, and W. H. Kwon, "Delay-dependent robust $\mathcal{H}_\infty$ control for uncertain systems with a state-delay," *Automatica*, vol. 40, no. 1, pp. 65–72, 2004.

[7] V. Balakrishnan, S. Boyd, and S. Balemi, "Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems," *International Journal of Robust and Nonlinear Control*, vol. 1, no. 4, pp. 295–317, October–December 1991.

[8] R. M'Closkey, A. Packard, and J. Spila, "Branch and bound computation of the minimum norm of a linear fractional transformation over a structured set," *IEEE Transactions on Automatic Control*, vol. 45, pp. 369–375, February 2000.

[9] K.-C. Goh, M. G. Safonov, and G. P. Papavassilopoulos, "Global optimization for the biaffine matriz inequality problem," *Journal of Global Optimization*, vol. 7, pp. 365–380, 1995.

[10] E. Assunção and P. L. D. Peres, "A global optimization approach for the $\mathcal{H}_2$-norm model reduction problem," in *Proceedings of the 38th Conference on Decision & Control*. Phoenix, Arizona USA: IEEE, December 1999, pp. 1857–1862.

[11] M. de Berg, M. V. Kreveld, M. Overmas, and O. Schwarzkopt, *Computational Geometry: Algorithms and Applications*. Springer Verlag, 1998.

[12] C. B. Barber, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, December 1996.

[13] H. Edelsbrunner and D. R. Grayson, "Edgewise subdivision of a simplex," *Discrete & Computational Geometry*, vol. 24, pp. 707–719, 2000.

[14] J. Bey, "Simplicial grid refinement: on Freudenthal's algorithm and the optimal number of congruence classes," *Numerische Mathematik*, vol. 85, pp. 1–29, 2000.

[15] C. E. de Souza and X. Li, "Delay-dependent robust $\mathcal{H}_\infty$ control of uncertain linear state-delayed systems," *Automatica*, vol. 35, no. 7, pp. 1313–1321, 1999.

[16] J. C. Geromel, M. C. de Oliveira, and J. Bernussou, "Robust filtering of discrete-time linear systems with parameter dependent lyapunov functions," *SIAM Journal of Control Optimization*, vol. 41, no. 3, pp. 700–711, 2002.