HAL
open science

A global optimization approach to structured regulation design under Hinfinity constraints

Dominique Monnet, Jordan Ninin, Benoit Clement

To cite this version:

Dominique Monnet, Jordan Ninin, Benoit Clement. A global optimization approach to structured regulation design under Hinfinity constraints. 55th IEEE Conference on Decision and Control CDC 2016), Dec 2016, Las Vegas, United States. hal-01416977

HAL Id: hal-01416977

https://hal.science/hal-01416977v1

Submitted on 15 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A global optimization approach to structured regulation design under H_∞ constraints.

Dominique Monnet, Jordan Ninin and Benoît Clément

***Abstract*— This paper presents a global optimization approach to the structured H_∞ sensitivity problem. The problem is formulated as a min/max optimization problem, and is solved with a branch and bound algorithm based on interval arithmetic. The method is compared with other existing H_∞ synthesis methods and results are discussed.**

I. Introduction

The H_∞ synthesis [20] is a method that formulates performance and robustness objectives as an optimization problem. The first methods developed to synthesize a controller from the H_∞ formulation [7], [6] met difficulties to reach a large audience since synthesized controllers cannot be implemented efficiently in practice. Indeed, these methods provide full order controllers which have to be truncated by order reduction or stucturation [17].

The synthesis of fixed order controllers, thus workable in real cases, is a challenging problem when meet robustness and performance constraints. The emergence of efficient methods to solve Linear Matrix Inequality problems (LMI) has encouraged formulating these constraints as LMI [4], [15], [8]. However, some problems cannot be cast into LMI and require other optimization methods to be solved, such as convex-concave optimization [9] or non-smooth optimization [5], [1].

Global optimization [11] approaches present a major interest for non-convex problems. Indeed, global optimization methods provide an enclosure of the global optimum of a problem. Furthermore, computation times of global optimization methods are no more prohibitive due to a lot of studies [14], [2] to improve the convergence of global optimization algorithms. We choose to approach the H_∞ sensitivity problem under a structural constraint on the controller with global optimization algorithm based on interval arithmetic to cope with the non-convexity of this problem.

This paper is organized as follows. Section II proposes a definition of the H_∞ problem and the H_∞ sensitivity problem. Section III introduces interval arithmetic and proposes an algorithm to solve the H_∞ sensitivity problem. Section IV compares our results to those obtained with two other H_∞ synthesis methods. Section V concludes this paper.

This work is supported by the city of Brest and the DGA (French Defense Procurement Agency).

D. Monnet is with Lab-STICC, ENSTA Bretagne, 2 rue François Verny, France dominique.monnet@ensta-bretagne.org

J. Ninin is with Lab-STICC, ENSTA Bretagne, 2 rue François Verny, France jordan.ninin@ensta-bretagne.fr

B. Clément is with Lab-STICC, ENSTA Bretagne, 2 rue François Verny, France benoit.clement@ensta-bretagne.fr

II. Structured synthesis under H_∞ constraints

In this section, we recall the H_∞ synthesis problem. Then, we show how performance and robustness constraints can be expressed as H_∞ constraints and we introduce the H_∞ sensitivity problem. Finally, we introduce the notion of structured controller and we formulate the problem we propose to solve.

A. H_∞ problem

The H_∞ synthesis problem can be defined as proposed in [19]. Given an Linear Time Invariant (LTI) system P and a set $\mathcal{K}$ of LTI systems, find a controller from $\mathcal{K}$ which is a solution to the problem:

$$\begin{array}{c}\underset{K\in\mathcal{K}}{\text{minimize}}\ ||F(P,K)||_\infty \\ \text{subject to } K \text{ internally stabilizes } P\end{array} \tag{1}$$

where $F(P,K)$ is the Linear Fractional Transform (LFT) that describes the closed loop system represented in Figure 1. In

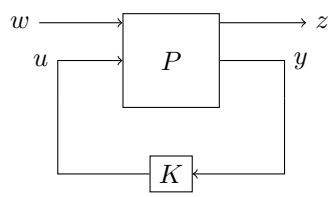

Fig. 1. Interconnected systems

Figure 1, w represents the external inputs, z the error or performance outputs, u the control inputs and y the measure outputs. $F(P,K)$ has the following frequency representation

$$F(P,K) = \begin{pmatrix} T_{w\to z_1}(s) \\ \vdots \\ T_{w\to z_p}(s) \end{pmatrix} \tag{2}$$

where $T_{w\to z_j}(s) = (T_{w_1\to z_j}(s), ..., T_{w_n\to z_j}(s))$. $T_{w_i\to z_j}(s)$ represents the channel from the ith external input to the jth error output. $s = i\omega$ denotes the Laplacian variable, where i is the imaginary unit and $\omega \geq 0$ the pulsation.

Problem 1 was solved with the DGKF algorithm [6] based on the resolution of Ricatti equations, and was later reformulated as a Linear Matrix Inequalities problem [7].

B. From H_∞ constraints to H_∞ problem

In practice, Problem 1 is formulated from H_∞ constraints. Consider a LTI system $G(s)$ and a controller $K(s)$ in a tracking error regulation loop as shown in Figure 2, where e represents the tracking error signal, u the control, y the measure, r the reference, d the disturbance and n the noise. A signal $\tilde{z_i} = W_i(s)z_i$ denotes the weighted counterpart of the output signal z_i.

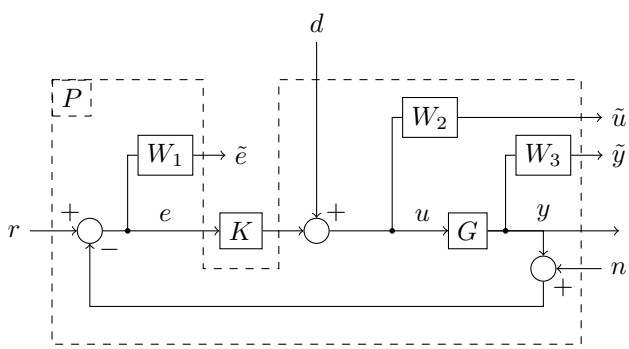

Fig. 2. H_∞ sensitivity approach.

The H_∞ norm of a LTI system that maps $w \in L_2$ into $z \in L_2$ represents the maximal gain of this system.

$$||T||_\infty = \max_{w \neq 0} \frac{||z||_2}{||w||_2} = \sup_\omega \sqrt{\lambda(T(i\omega)T(i\omega)^*)}, \quad (3)$$

where λ is the greatest eigenvalue and $T(i\omega)^*$ is the hermitian transpose of $T(i\omega)$. Thus, minimizing $||T||_\infty$ means minimizing the maximal response of the system to inputs w. The principle of H_∞ synthesis is to minimize infinity norm of objective channels $T_{w\to\tilde{z_i}}(s)$. Error outputs e, u and y are weighted with rational transfer functions $W_1(s)$, $W_2(s)$ and $W_3(s)$, such as non-desired behaviors of the closed loop system are penalized. For example, a non-desired behavior may be an important tracking error signal e at low frequencies (steady state). In this case, $W_1(s)$ penalizes the sensitivity function $T_{w\to e}(k,s)$ at low frequencies but not at high frequencies where noise n occurs. An example of robust constraint would be to minimize the action of actuators at high frequencies. Thus, $W_2(s)$ would penalize $T_{w\to u}(s)$ at these frequencies.

Once weighting functions $W_i(s)$ are defined, the objective is to find $K(s)$ that minimizes every objectives $||T_{w\to\tilde{z_i}}(s)||_\infty$ all at once, so that the closed loop system offers the desired behavior. In addition, $K(s)$ must internally stabilize the closed loop system.

In practice, H_∞ performance and robustness constraints are expressed as bounds on the gain of objective channels. These bounds are frequency templates designed from user specifications, and represent the inverse of weighting functions $W_i(s)$:

$$\begin{aligned} \forall\omega,\ \sqrt{\lambda(T_{w\to z_i}(i\omega)T_{w\to z_i}(i\omega)^*)} &\leq |W_i^{-1}(s)| \\ \iff ||T_{w\to z_i}(s)W_i(s)||_\infty &\leq 1 \\ \iff ||T_{w\to\tilde{z_i}}(s)||_\infty &\leq 1 \end{aligned} \quad (4)$$

H_∞ constraints can be interpreted as a frequency dependent upper bound on the ratio between an error output and inputs. Ensuring that every H_∞ constraints are respected can be formulated as a Constraints Satisfaction Problem (CSP),

$$\text{find } K, \text{ subject to } \begin{cases} ||T_{w\to\tilde{z_1}}(s)||_\infty \leq 1 \\ \vdots \\ ||T_{w\to\tilde{z_p}}(s)||_\infty \leq 1 \\ K(s) \text{ internally stabilizes } P \end{cases} \quad (5)$$

This CSP problem can be considered as an optimization problem, that we call the H_∞ sensitivity problem:

$$\begin{aligned} \underset{K}{\text{minimize}}\ & \max(||T_{w\to\tilde{z_1}}(s)||_\infty, ..., ||T_{w\to\tilde{z_p}}(s)||_\infty) \\ & \text{subject to } K(s) \text{ internally stabilizes } P \end{aligned} \quad (6)$$

Indeed, if the minimum of Problem 6 is lower than one, the solution of Problem 6 is a feasible solution to Problem 5. The classical way to compute a solution to Problem 5 is to solve the H_∞ problem. The system $P(s)$ is built from $G(s)$ and the weighting functions $W_i(s)$ as shown in Figure 2. Then, a solution to Problem 5 can be computed by solving Problem 9. If $||F(P,K)||_\infty$ is lower than one, then K is a solution to Problem 5. Indeed,

$$\begin{aligned} ||F(P,K)||_\infty &= ||\begin{pmatrix} T_{w\to\tilde{z_1}}(s) \\ \vdots \\ T_{w\to\tilde{z_p}}(s) \end{pmatrix}||_\infty \\ &\geq \max(||T_{w\to\tilde{z_1}}(s)||_\infty, ..., ||T_{w\to\tilde{z_p}}(s)||_\infty). \end{aligned} \quad (7)$$

For example, we can use [6] or [7] to solve the H_∞ problem. However, these methods give solutions in the space of full-order controllers, which means that the order of the controller is equal to the one of $P(s)$. Moreover, the more H_∞ constraints must be respected, the higher is the order of $P(s)$. Therefore, controllers computed by these two methods are usually high orders.

Remark 1: Due to Equation (7), if the minimum of Problem 9 is greater than one, it does not imply that Problem 5 has no solution.

C. Structured H_∞ problem

In practice low-order controllers, possibly mixed with filters, are preferred because their dynamics can be interpreted easily. A structured controller depends on free parameters $k \in \mathbb{K}$, where $\mathbb{K}$ denotes the space of controller parameters. For example, a PID controller depends on k_p, k_i and k_d.

$$K(k,s) = k_p + \frac{k_i}{s} + k_d s, \text{ with } k = (k_p, k_i, k_d) \quad (8)$$

As $F(P,K)$ represents the system $P(s)$ interconnected with $K(k,s)$, its elements $T_{w_i \to \tilde{z_j}}(k,s)$ depend on k. The structured H_∞ problem can be formulated as follows:

$$\begin{aligned} &\underset{k \in \mathbb{K}}{\text{minimize}} \ ||F(P,K)||_\infty \\ &\text{subject to } K(k,s) \text{ internally stabilizes } P \end{aligned} \tag{9}$$

One method to solve this problem is the non-smooth synthesis [1].
However, to our knowledge , no methods have been proposed to solve directly Problem 6 under a structural constraint on the controller. This problem can be expressed as follows,

$$\begin{aligned} &\underset{k \in \mathbb{K}}{\text{minimize}} \ \max(||T_{w \to \tilde{z_1}}(k,s)||_\infty, ..., ||T_{w \to \tilde{z_p}}(k,s)||_\infty) \\ &\text{subject to } K(k,s) \text{ internally stabilizes } P \end{aligned} \tag{10}$$

We propose to solve Problem 10 using a global optimization approach based on interval arithmetic.

III. Global optimization approach based on interval arithmetic

In this section, we show that Problem 10 can be solved in a guaranteed way using global optimization based on interval arithmetic. Moreover, a global optimization approach enables to compute an enclosure of the minimum of the objective function. Thus a certificate of infeasibility of Problem 5 can be obtained.
We first introduce interval arithmetic and the notion of inclusion function [10]. Then, we reformulate Problem 10 as a min/max problem. After that we show how to tackle the maximization part of this min/max problem with interval arithmetic. Finally, we propose a branch and bound algorithm to solve Problem 10.

A. Interval Arithmetic

An Interval is a closed connected subset of $\mathbb{R}$ [10]. Intervals are denoted using boldface letters $\boldsymbol{x}$. A non-empty interval $\boldsymbol{x}$ can be represented by its endpoints:

$$\boldsymbol{x} = [\underline{x}, \overline{x}] = \{x : \underline{x} \leq x \leq \overline{x}\}$$

with $\underline{x} \in \mathbb{R} \cup \{-\infty\}$, $\overline{x} \in \mathbb{R} \cup \{+\infty\}$ and $\underline{x} \leq \overline{x}$. The set of intervals is denoted by $\mathbb{IR}$ and the set of n-dimensional interval vectors, also called boxes, is denoted by $\mathbb{IR}^n$.
The main advantage of interval arithmetic is the reliability of computations, in the sense that a guaranteed enclosure of the result is computed. This inclusion property is the fundamental theorem of interval arithmetic. Let $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ be a function. An inclusion function $\boldsymbol{g} : \mathbb{IR}^n \mapsto \mathbb{IR}^m$ of g is defined as follows,

$$\forall \boldsymbol{x} \in \mathbb{IR}^n, g(\boldsymbol{x}) = \{g(x), x \in \boldsymbol{x}\} \subseteq \boldsymbol{g}(\boldsymbol{x}) \tag{11}$$

An inclusion function provides an upper and a lower bound on a function over an interval. Several techniques can be used to construct an inclusion function of every factorable function involving $+, -, *, sin, exp, \ max, min, ...$ [16]. Figure 3 illustrates the concept of inclusion function. It is important to note that $\boldsymbol{g}(\boldsymbol{x})$ in light gray in Figure 3 is an over approximation of $g(\boldsymbol{x})$ displayed in dark gray, and is not the smallest possible box which encloses $g(\boldsymbol{x})$.

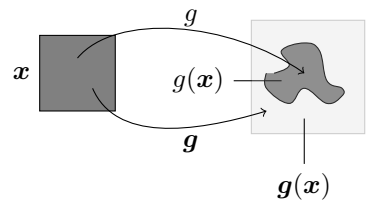

Fig. 3. Inclusion function

B. Interval approach of the problem

The computation of $||.||_\infty$ involves dealing with singular values (see Equation (3)). Unfortunately, an interval arithmetic approach does not allow to handle efficiently the computation of singular values. However, it is possible to avoid this difficulty by reformulating Problem 10. Indeed, $||T_{w \to z_j}(k,s)||_\infty$ can be formulated as the maximization of a scalar-valued function:

$$\begin{aligned} ||T_{w \to z_j}(k,s)||_\infty &= \sup_\omega \sqrt{\lambda(T_{w \to \tilde{z_j}}(k,i\omega) T_{w \to \tilde{z_j}}(k,i\omega)^*)} \\ &= \sup_\omega \sqrt{\lambda(T_{w_1 \to \tilde{z_j}}(k,s), ...T_{w_n \to \tilde{z_j}}(k,s)) \begin{pmatrix} \overline{T_{w_1 \to \tilde{z_j}}(k,s)} \\ \vdots \\ \overline{T_{w_n \to \tilde{z_j}}(k,s)} \end{pmatrix}} \\ &= \sup_\omega \sqrt{\sum_{i=1}^n |T_{w_i \to \tilde{z_j}}(k,s)|^2} \end{aligned} \tag{12}$$

Then, Problem 10 can be reformulated as follows:

$$\begin{aligned} &\underset{k}{\text{minimize}} \ \sup_\omega f(k,\omega) \\ &\text{subject to } K \text{ internally stabilizes } P \end{aligned} \tag{13}$$

where

$$\begin{aligned} f(k,\omega) = \max(&\sqrt{\sum_{i=1}^n |T_{w_i \to \tilde{z_1}}(i\omega,k)|^2}, ..., \\ &\sqrt{\sum_{i=1}^n |T_{w_i \to \tilde{z_p}}(i\omega,k)|^2}). \end{aligned} \tag{14}$$

The internal stability constraint is generally expressed as a LMI [3]. Such a constraint cannot be mixed properly with a global optimization approach based on interval arithmetic. However, the stability constraint can be dealt with interval arithmetic using the Routh criterion on the characteristic polynomial of the state matrix of $F(P,K)$ [18]. Indeed, the constraint "$K(k,i\omega)$ internally stabilizes P" can be reformulated as a set of non-convex constraints on k, denoted $\mathcal{R}(k) \leq 0$. The loop shaping problem is formulated as a constrained min/max problem that can be solved using a branch and bound algorithm based on interval arithmetic.

C. Enclosure of the objective function

The objective function of Problem 13 involves the computation of $f_{sup}(k) = \sup_\omega f(k,\omega)$. To minimize this objective

function using a branch and bound algorithm, we need to compute a lower and an upper bound of f_{sup} over a box $\boldsymbol{k}$. This bound must be as close as possible to the exact range of f_{sup} to ensure the convergence of the branch and bound algorithm.

Using interval arithmetic, an inclusion function $\boldsymbol{f}$ of $f(k,\omega)$ can be built. Indeed, the operators $+$, $*$, abs, $\sqrt{\ }$, max are well-defined [11]. In order to use interval arithmetic, we limit the study of $\sup_{\omega} f(k,\omega)$ to a bounded set Ω. Thus, we obtain the following equation:

$$\forall \boldsymbol{k} \in \mathbb{IR}^n, f_{sup}(\boldsymbol{k}) = \left\{ \sup_{\omega \in \Omega} f(k,\omega) \ : \ k \in \boldsymbol{k} \right\} \subseteq \boldsymbol{f}(\boldsymbol{k},\Omega).$$

Unfortunately, the bounds obtained using the inclusion function directly over Ω are not close enough to $f_{sup}(\boldsymbol{k})$. That is why we subdivide Ω into several boxes $\boldsymbol{\omega}_i$ such as $\Omega = \bigcup_i \boldsymbol{\omega}_i$. Thus, we have the following result:

$$\forall \boldsymbol{k} \in \mathbb{IR}^n, f_{sup}(\boldsymbol{k}) \subseteq \left[\max_i \underline{\boldsymbol{f}(\boldsymbol{k},\boldsymbol{\omega}_i)}, \max_i \overline{\boldsymbol{f}(\boldsymbol{k},\boldsymbol{\omega}_i)} \right].$$

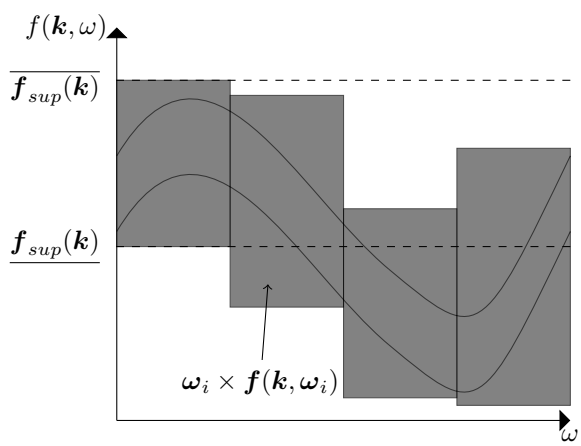

Fig. 4. Computation of $\boldsymbol{f}_{sup}(\boldsymbol{k})$.

Figure 4 illustrates the previous equation. Ω is discretized with non-overlapping intervals and $\boldsymbol{f}$ is evaluated for each of them. The highest lower bound and upper bound of the evaluations represented by dotted lines in Figure 4 give a guaranteed enclosure of $f_{sup}(\boldsymbol{k})$. Indeed, the function $\boldsymbol{f}_{sup}$, defined as follows, is an inclusion function of f_{sup}:

$$\forall \boldsymbol{k} \in \mathbb{IR}^n, \boldsymbol{f}_{sup}(\boldsymbol{k}) = \left[\max_i \underline{\boldsymbol{f}(\boldsymbol{k},\boldsymbol{\omega}_i)}, \max_i \overline{\boldsymbol{f}(\boldsymbol{k},\boldsymbol{\omega}_i)} \right].$$

D. Global Optimization Algorithm

We propose to solve Problem 13 with a branch and bound algorithm [11], [13]. We denote μ the global minimum of Problem 13, $\mathbf{K} \subseteq \mathbb{K}$ the initial domain of controller variables, lb_μ and ub_μ a lower and an upper bound of μ.

The principle of a branch and bound algorithm is to split $\mathbf{K}$ into smaller subsets and eliminate them if they do not satisfy the stability constraint or if it is certified that they do not contain the global solution. When the algorithm ends, we obtain:

- k^* the best feasible solution found,
- $[lb_\mu, ub_\mu]$ a reliable enclosure of the minimum μ,
- A certificate of infeasibility to Problem 5 if no feasible solution has been found or $lb_\mu > 1$.

1: **while** $|ub_\mu - lb_\mu| \geq \epsilon$ and $\mathcal{L} \neq \emptyset$ **do**
2: Extract a box $\boldsymbol{k}$ from $\mathcal{L}$
3: Contract $\boldsymbol{k}$ using [14].
4: Bissect $\boldsymbol{k}$ into $\boldsymbol{k_1}$ and $\boldsymbol{k_2}$.
5: **for** $i = 1, 2$ **do**
6: Compute $\boldsymbol{f}_{sup}(\boldsymbol{k_i})$.
7: **if** $\underline{\boldsymbol{f}_{sup}(\boldsymbol{k}_i)} \leq ub_\mu$ **then**
8: Add $\boldsymbol{k_i}$ in $\mathcal{L}$.
9: **end if**
10: Choose $k \in \boldsymbol{k}_i$ that respects $\mathcal{R}(k) \leq 0$.
11: **if** $\overline{\boldsymbol{f}_{sup}(k)} < ub_\mu$ **then**
12: Update ub_μ and $k^* = k$.
13: **end if**
14: **end for**
15: Update $lb_\mu = \min_{\boldsymbol{k} \in \mathcal{L}} \underline{\boldsymbol{f}_{sup}(\boldsymbol{k})}$.
16: **end while**

Algorithm 1: Branch and Bound algorithm.

Algorithm 1 works as follows. A list of boxes $\mathcal{L}$ is initialized with $\mathbf{K}$. At Line 2, the box $\boldsymbol{k}$ with the lowest lower bound $\underline{\boldsymbol{f}_{sup}(\boldsymbol{k})}$ is chosen from $\mathcal{L}$. At Line 3, $\boldsymbol{k}$ is contracted on its feasible region using [14]. If the result is not empty, $\boldsymbol{k}$ is bisected in two non-overlapping boxes $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$. For $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$, $\boldsymbol{f}_{sup}(\boldsymbol{k}_i)$ is computed at Line 6 using Section III-C. If $\underline{\boldsymbol{f}_{sup}(\boldsymbol{k})} > ub_\mu$, it is proved that the global minimum cannot belong to $\boldsymbol{k}$, else the box is added to $\mathcal{L}$. At Line 10, if possible, a point k that stabilizes $P(s)$ is chosen in $\boldsymbol{k}$. If $\overline{\boldsymbol{f}_{sup}(k)}$ is lower than the current value ub_μ, k is a better solution and ub_μ is updated.
When Algorithm 1 stops, $\mathcal{L}$ contains a set of boxes that contain the global minimizer; $[lb_\mu, ub_\mu]$ provides a guaranteed enclosure of the global minimum μ; k^* is the best known solution. If $(lb_\mu > 1)$ or if $(\mathcal{L} = \emptyset) \wedge (k^* = \emptyset)$, it is certified that Problem 5 has no solution.

IV. Example

The methodology is applied on a simple example and compared with methods presented in [1] and [7]. We use the Matlab Robust Control toolbox where both of them are implemented.

A. Problem and results

We consider the second order LTI system:

$$G(s) = \frac{1}{s^2 + 1.4s + 1},$$

and a PID with a derivative filtering:

$$K(k,s) = k_p + \frac{k_i}{s} + \frac{k_d s}{1+s}.$$

The vector of controller parameters is $k = (k_p, k_i, k_d)$. Each parameter belongs to $[-10, 10]$, then $\mathbf{K} = [-10, 10]^3$ and the frequency domain is $\Omega = [10^{-3}, 10^3]$. We consider the regulation scheme of Figure 2 without input signals d and n. It means that we have one input signal r and three weighted outputs $\tilde{z}_1$, $\tilde{z}_2$ and $\tilde{z}_3$. Weighting functions are the following ones

$$W_1(s) = \frac{10s+100}{1000s+1}, \quad W_2(s) = \frac{10s+1}{s+10}, \quad W_3(s) = \frac{100s+1}{s+10}. \tag{15}$$

The templates of their inverses appear as dotted lines in Figure 5.

We compute a controller that is a solution of Problem 5. To do so, we solve Problem 1 with [7], Problem 9 with [1] and Problem 10 with a global optimization approach. These methods are denoted respectively by H_∞ full, H_∞ struct and Global Optimization (GO) struct in the first column of Table I that presents the results obtained.

The stop criterion of GO struct method is $\epsilon = 0.1$. The

Method	Cpu (s)	$\|\|F(P,K)\|\|_\infty$	$\max_i(\|\|T_{w\to\tilde{z}_i}\|\|_\infty)$
H_∞ full	2	1.0258	1.01611
H_∞ struct	80	1.0411	1.04109
GO struct	83	1.0811	0.99782

TABLE I
RESULTS OBTAINED.

second column indicates computation times in second. The H_∞ norm of $P(s)$ interconnected with the controller appears in the third column (see Figure 2). The value of the objective function of the sensitivity problem is indicated in the last column. The H_∞ full method gives a 5th order controller denoted $K_{H_\infty full}$, H_∞ struct provides the following solution:

$$K_{H_\infty struct}(k,s) = 0.0736 + \frac{0.0969}{s} + \frac{0.0305s}{1+s}$$

with our algorithm, we obtain:

$$K_{GO}(k,s) = 0.0348 + \frac{0.0993}{s} + \frac{0.0625s}{1+s}$$

Furthermore, our method gives the following enclosure, $\max_i(||T_{w\to\tilde{z}_i}||_\infty) \in [0.905531, 0.997827]$.

In the sequel, $S_{H_\infty full} = (I + G(s)K_{H_\infty full}(s))^{-1}$ denotes the sensitivity function defined with $K_{H_\infty full}$, $S_{H_\infty struct}$ and S_{go} are defined the same way replacing $K_{H_\infty full}$ by $K_{H_\infty struct}$ and K_{GO} in the expression. Similarly, we defined the complementary sensitivity functions $T_{H_\infty full} = G(s)K_{H_\infty full}(s)(I + G(s)K_{H_\infty full}(s))^{-1}$, $T_{H_\infty struct}(s)$ and $T_{go}(s)$.

B. Discussion

H_∞ full method gives a result rapidly due to the convexity of the problem. As H_∞ structured method is based on local optimization, we run it 350 times with random start points such that the computation time is close to the one of GO structured. Considering $||F(P,K)||_\infty$ criterion, H_∞ full gives a better result than H_∞ struct because $K_{H_\infty full}$ has a higher order. GO struct gives the worst result, but this method minimizes $\max_i(||T_{w\to\tilde{z}_i}||_\infty)$ instead of $||F(P,K)||_\infty$ contrary to the other two methods. However, if we look at $\max_i(||T_{w\to\tilde{z}_i}||_\infty)$ criterion, one can remark that only the GO method is able to compute a solution lower than 1, *i.e.* to provide a controller that respects all the three frequency constraints $||T_{w\to\tilde{z}_i}(s,k)||_\infty \leq 1, i \in \{1,2,3\}$. This can be explained by the fact that both $K(k,s)$ and $G(s)$ act like low-pass filters and their outputs u and y are both weighted. This leads to coupled terms in $P(s)$, thus $P(s)P(s)^*$ has got non-negligible terms outside its diagonal. Therefore, minimizing $||P(s)||_\infty$ is a different problem than minimizing $\max_i(||T_{w\to\tilde{z}_i}||_\infty)$. However, $\max_i(||T_{w\to\tilde{z}_i}||_\infty)$ value of the three methods are close to each other. This is illustrated by Figure 5. Bode diagrams of the objective channels given by

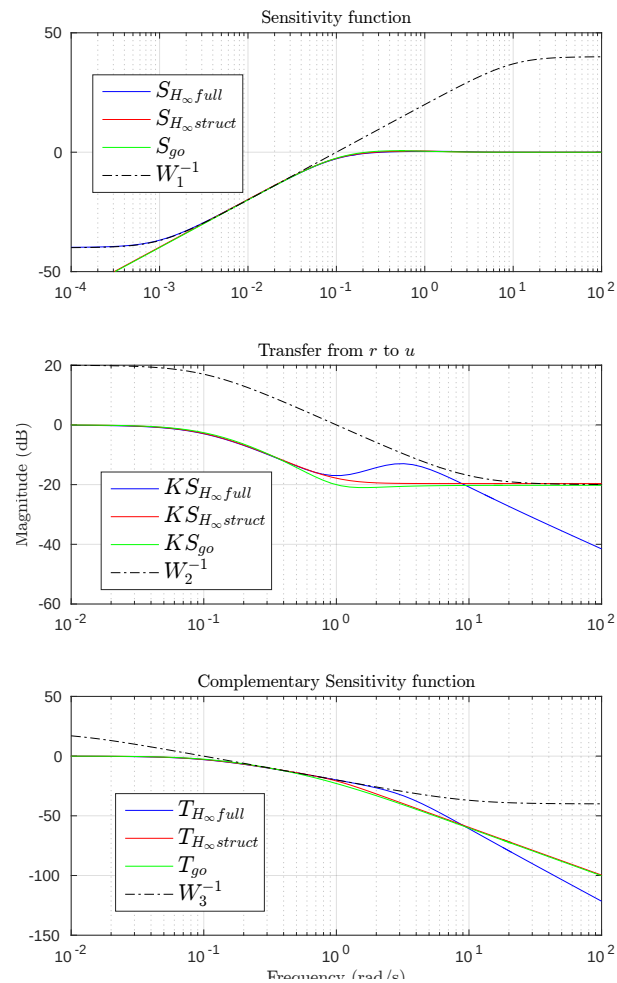

Fig. 5. Respect of frequency constraints.

the three methods are close to the frequency templates of the weighting functions. One can see that S_{full} is above W_1^{-1} at low frequencies and KS_{struct} is above W_2^{-1} at high frequencies, which illustrates that $\max_i(||T_{w\to\tilde{z}_i}||_\infty)$ obtained with H_∞ full and H_∞ struct are greater than 1.

Although the proportional and derivative coefficients of controllers computed by H_∞ struct and GO struct methods present significant differences, their Bode diagrams, shown in Figure 6, are almost the same. The bode diagram of the

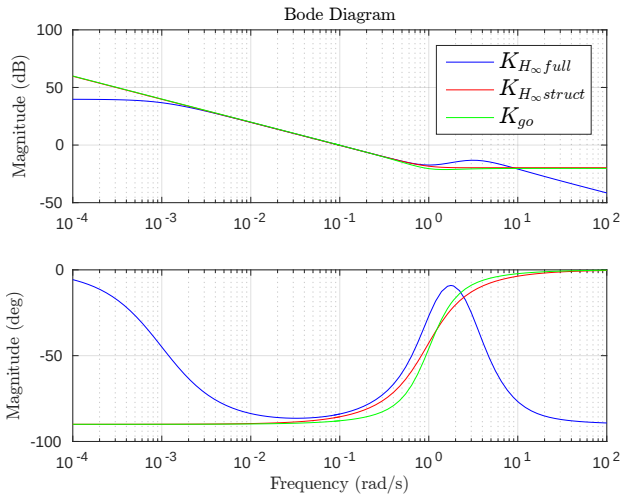

Fig. 6. Bode diagrams of controllers.

controller computed by H_∞ full shows a more complex behavior due to the high order.

Despite the fact that controllers obtained with the three different methods present almost similar performances, this example shed a light on the fact that GO approach is not sensitive to coupled constraints, contrary to classical approaches which aim to minimize $||P(s)||_\infty$.

In this example, we chose fractional weighting functions $W_i(s)$ to compare our method with H_∞ full and H_∞ struct. However GO method can be extended to the case where the weighting functions $W_i(s)$ are not rational, provided that we have an inclusion function of f. One can remark that if weighting functions are not rational, $P(s)$ is not linear anymore and the internal stability cannot be ensure with the Routh criterion. Nonetheless, internal stability of $P(s)$ is not needed in practice, it suffices that $K(s)$ internally stabilizes the closed loop system (*i.e.* without weighting functions on objective channels).

Furthermore, an interval arithmetic approach allows to deal with parametric uncertainties. Indeed, if $G(\boldsymbol{p}, s)$ depends on uncertain parameters $\boldsymbol{p}$ defined as intervals, it is still possible to compute $\boldsymbol{f}(\boldsymbol{k}, \boldsymbol{p}, s)$ and therefore to run Algorithm 1. Ensuring a performance constraint with parametric uncertainties has already been investigated with interval arithmetic for SISO system [12], but our approach is more general and can be applied to MIMO systems.

V. Conclusion

In this paper, we have introduced a method based on global optimization to solve the sensitivity problem given a structured controller. We show how to compute a guaranteed enclosure of the minimum of the loop shaping problem. Thus, it is possible to provide a certificate of infeasibility of the sub-optimal problem. We illustrated the advantage of a global optimization method compared to two classical approaches with an academic example. Indeed, our approach considers each weighted outputs separately so that our synthesis method is not sensitive to coupled constraints.

Moreover, the approach proposed in this paper offers new prospects. It is possible to take parameters uncertainties into account without modification of the method. In addition, our method does not restrict the expression of weighting functions as rational ones.

Even if the computation time can be improved, the proposed approach opens up new ways to synthesize and analyze robust controllers.

References

[1] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *Automatic Control, IEEE Transactions on*, 51(1):71–86, 2006.

[2] R. E. Bixby. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, Extra Volume: Optimization Stories:107–121, 2012.

[3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, 1994.

[4] S. Boyd, M. Hast, and K. J. Åström. MIMO PID tuning via iterated LMI restriction. 2015.

[5] JV Burke, D Henrion, AS Lewis, and ML Overton. Hifoo-a matlab package for fixed-order controller design and H_∞ optimization. 2006.

[6] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard h2 and H_∞ control problems. *Automatic Control, IEEE Transactions on*, 34(8):831–847, 1989.

[7] P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *International journal of robust and nonlinear control*, 4(4):421–448, 1994.

[8] O. Garpinger and T. Hägglund. A software tool for robust PID design. *Proc. 17th IFAC World Congress, Seoul, Korea*, 2008.

[9] M. Hast, K. J. Aström, B. Bernhardsson, and S. Boyd. PID design by convex-concave optimization. *Proceedings European Control Conference*, pages 4460–4465, 2013.

[10] Luc Jaulin, Michel Kieffer, Olivier Didrit, and Éric Walter. *Applied Interval Analysis*. Springer London, 2001.

[11] R. B. Kearfott. An interval branch and bound algorithm for bound constrained optimization problems. *Journal of Global Optimization*, 2(3):259–280, 1992.

[12] S. Malan, M. Milanese, and M. Taragna. Robust analysis and design of control systems using interval arithmetic. *Automatica*, 33(7):1363–1372, 1997.

[13] J. Ninin. *Global Optimization based on Interval Analysis: Affine Relaxation and Limited Memory*. Theses, Institut National Polytechnique de Toulouse - INPT, December 2010.

[14] J. Ninin, F. Messine, and P. Hansen. A reliable affine relaxation method for global optimization. *4OR: A Quarterly Journal of Operations Research*, 13(3):247–277, September 2015.

[15] M. Sadeghpour, V. de Oliveira, and A. Karimi. A toolbox for robust PID controller tuning using convex optimization. *Proceedings of IFAC Conference in Advances in PID Control*, 2012.

[16] B. Tóth. *Interval Method for Competitive Location Problems*. phdthesis, 2007.

[17] O. Voinot, D. Alazard, and B. Clément. Unstationnary control of a launcher using observer-based structures. In *Proceedings of American Control Conference*, Denver, USA, 2003.

[18] E. Walter and L. Jaulin. Guaranteed characterization of stability domains via set inversion. *IEEE Transactions on Automatic Control*, 39(4):886–889, 1994.

[19] G. Zames. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26(2):301–320, 1981.

[20] K. Zhou and J. C. Doyle. *Essentials of Robust Control*. Prentice Hall, 1998.