Multiobjective synthesis using LMI formulations for application of the cutting plane algorithm

Mohamed Abbas-Turki[1], Gilles Duc[1*], Benoît Clement[2]

[1]Supélec, Département d'Automatique

3, rue Joliot Curie, 91192 Gif-sur-Yvette, France

{mohamed.abbas-turki, gilles.duc}@supelec.fr

[2] CNES, Direction des Lanceurs

Rond Point de l'Espace, 91023 Evry cedex, France

benoit.clement@cnes.fr

phone: +33 1 69 85 13 88

fax: +33 1 69 85 13 89

[*]Corresponding author

Abstract

The problem of designing a controller to meet different specifications or deciding that no such controller exists is addressed in this paper by linking three types of tools: the Youla parameterization allows searching for a controller in a convex set; formulations using Linear Matrix Inequalities (LMI) are proposed for different practical specifications; the corresponding convex problem is solved using a Cutting Plane Algorithm (CPA). Such an approach is developed by overcoming the problem of the huge number of additional variables which often occurs in the LMI framework, particulary when used in conjunction with the Youla parameterization. Its efficiency is discussed by considering two practical control problems.

Key words: multiobjective control, Youla parameterization, LMI, Cutting Plane Algorithm, KYP lemma.

1 Introduction

This paper considers the problem of designing a controller according to different practical specifications or to be sure that no such controller exists. It can be included in the class of multiobjective control problems, which have been the subject of several works (see for instance [3, 15, 11, 13, 18, 5]). In fact, a lot of questions have to be addressed, including:

- the translation of the physical specifications into an appropriate mathematical form;
- the formulation of the multiobjective problem as a convex one either by finding an appropriate transformation or by doing some relaxation to recover the convexity; [1]
- the numerical feasibility (i.e. the amount of time calculation and memory space involved).

[1] if the problem is non convex, which occurs in most cases, a suitable algorithm has to be developed, so that the resolution becomes specific to the problem considered.

The first point can be provided by the Linear Matrix Inequalities (LMI) framework [4]. This formulation is attractive because LMIs describe a convex set and can be solved using powerful numerical techniques. However, if LMI formulations are well known for $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norm constraints or for pole placement for instance [15], such constraints are only indirect translations of the specifications in most cases (for instance, pole placement handles imperfectly a requirement on the settling time).

The Youla parameterization [12, 3] is useful to handle the second point: it defines a convex set describing all stabilizing controllers, so that no possible solution is lost. Unfortunately using the Youla parameter can lead to lose the linearity of the constraints, unless some particular transformations are used, as for instance in [16, 8]. Furthermore, to obtain a finite dimensional problem, a truncated projection of the Youla parameter on a given basis is usually looked for.

On another hand, this approach provides a huge controller order, especially if the constraints are harsh. Moreover, most commonly used LMI formulations generally require introducing matrices of the same order as the closed-loop plant, so that the third point is not always met. In order to avoid such additional variables, Kao [9] presents a method based on the eigenvalues of some Hamiltonian matrix, and the application of a Cutting Plane Algorithm (CPA) instead of using Semi-Definite Programming (SDP). Although this method is more sensitive to numerical conditioning, it is less affected by the order of the plant.

The purpose of the paper is therefore to propose a practical methodology by linking these tools together: by describing the controller using a truncation of the Youla parameter, different useful criteria (such as time response shaping, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norm constraints, roll-off specifications, stability margins) will be translated on LMI formulations which are suitable for applying the CPA. The efficiency of this approach will be discussed by considering practical problems and comparing with solutions obtained using a classical SDP solver.

It is organized as follows: section 2 contains a brief summary of the Youla parameterization; section 3 explains how the CPA can solve a convex feasibility problem; in section 4, LMI formulations of different criteria close to the practical specifications are derived. The application of the proposed methodology is the subject of section 5, where two control problems are discussed.

2 Youla parameterization

2.1 Parameterization of the set of stabilizing controllers

Consider a continuous or discrete-time plant G, with state-space realization:

$$G = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} : \quad \begin{matrix} & w & u \\ z & & \\ y & & \end{matrix} \left(\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right) \tag{1}$$

where $z \in \mathbb{R}^{m_1}$ is the output to be controlled despite disturbance $w \in \mathbb{R}^{p_1}$, using control input $u \in \mathbb{R}^{p_2}$ and measurement $y \in \mathbb{R}^{m_2}$. All stabilizing controllers for G are described by the Redheffer product $K = J * Q$ (see the interconnection structure of figure 1), where the Youla parameter Q is any stable transfer function [3]. System J depends both on stable coprime factorizations of G_{22} and an initial compensator K_0.

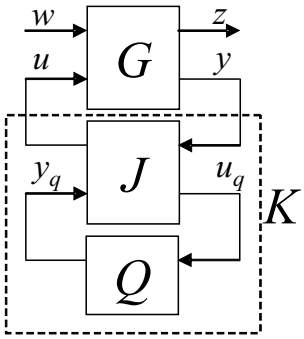


Figure 1: Closed-loop structure using Youla parameterization

As a main result, the closed-loop transfer G_{zw} depends linearly on Q:

$$G_{zw} = \left(G_{11} + G_{12}U_0\tilde{M}G_{21}\right) + \left(G_{12}M\right)Q\left(\tilde{M}G_{21}\right) = H_{11} + H_{12}QH_{21} \tag{2}$$

In the following, a state-space representation of G_{zw} will be noted by $(A_{zw}, B_{zw}, C_{zw}, D_{zw})$.

2.2 Finite dimensional approximation of the Youla parameter

To obtain a finite dimensional controller, the Youla parameter is approximated by truncating its projection on a given basis [8, 16]. For MIMO models, such an approximation can be written:

$$Q(\varsigma) = \sum_{j=1}^{m_2} \sum_{u=1}^{p_2} \left(\sum_{k=0}^{n_q} q_{k,j,u} Q_{k,j,u}(\varsigma) \right) e_j e_u^T = \sum_{m_2,p_2} Q_{j,u}(\varsigma) e_j e_u^T \tag{3}$$

where e_i is the unitary vector whose i-th element is equal to 1, $Q_{j,u}$ is SISO and υ is either the discrete-time or Laplace operator. $\{Q_{k,j,u}\}$ is a chosen basis of stable transfer functions and $q_{k,j,u}$ are the design parameters. Let $(A_{Q_{j,u}}, B_{Q_{j,u}}, C_{Q_{j,u}}, D_{Q_{j,u}})$ be a state-space realization of $Q_{j,u}$: matrices $A_{Q_{j,u}}$ and $B_{Q_{j,u}}$ are fixed by the choice of $\{Q_{k,j,u}\}$, so that all the design parameters $q_{k,j,u}$ enter in $C_{Q_{j,u}}$ and $D_{Q_{j,u}}$ only.

It remains to put the design variables in C_{zw} and D_{zw} only, for guaranteing in most cases the linearity of the matrix inequalities constraints with respect to the design parameters. A suitable technique has been proposed by [8] using the Kronecker product. Taking (3) on G_{zw} given by (2) leads to:

$$\begin{aligned} G_{zw} &= H_{11} + H_{12} \left(\sum_{m_2,p_2} Q_{j,u} e_j e_u^T \right) H_{21} \\ &= H_{11} + \sum_{m_2,p_2} Q_{j,u} \otimes T_{j,u} \end{aligned} \tag{4}$$

with: $T_{j,u} = (H_{12} e_j) \left(e_u^T H_{21} \right)$.

Let $\left(A_{T_{j,u}}, B_{T_{j,u}}, C_{T_{j,u}}, D_{T_{j,u}} \right)$ be state-space representations of $T_{j,u}$ for each values of j, u respectively. The Kronecker product $Q_{j,u} \otimes T_{j,u}$ has the following state-space realization:

$$\left(\begin{array}{c|c} A_{Q_{j,u} \otimes T_{j,u}} & B_{Q_{j,u} \otimes T_{j,u}} \\ \hline C_{Q_{j,u} \otimes T_{j,u}} & D_{Q_{j,u} \otimes T_{j,u}} \end{array} \right) = \left(\begin{array}{cc|c} A_{Q_{j,u}} \otimes I_{m_1} & B_{Q_{j,u}} \otimes C_{T_{j,u}} & B_{Q_{j,u}} \otimes D_{T_{j,u}} \\ 0 & A_{T_{j,u}} & B_{T_{j,u}} \\ \hline C_{Q_{j,u}} \otimes I_{m_1} & D_{Q_{j,u}} \otimes C_{T_{j,u}} & D_{Q_{j,u}} \otimes D_{T_{j,u}} \end{array} \right) \tag{5}$$

As it can be noticed from (5), all design variables enter only in matrices $C_{Q_{j,u}\otimes T_{j,u}}$, $D_{Q_{j,u}\otimes T_{j,u}}$ and consequently, from (4), they appear only in matrices C_{zw} and D_{zw}.

This representation leads to a non-minimal state space representation of G_{zw} having a high order, particularly for MIMO plants (that is $n + 2\,n\,m\,p_2 + 2\,m\,p_1\,n_Q\,m_1$, where n and n_Q are respectively the dimensions of matrices A, A_Q). This means that for avoiding numerical infeasibility, one has to search for a synthesis method which is less sensitive to the state-space order: for this reason, all methods based on introducing a matrix having the same order as A_{zw} should be avoided.

Remark 1 *Scherer [16] presented another technique, where a minimal form of G_{zw} is used but suitable transformations are required to restore the linearity of the constraints. Note also that the transformation is done for $\mathcal{H}_\infty$ and $\mathcal{H}_2$ norm constraints, but does not apply for some of the constraints that are considered below.*

3 The Cutting Plane Algorithm

This section presents a variant of the Cutting Plane Algorithm (CPA) presented in [9], where the convergence for a convex problem is guaranteed [19]. Only the case of a feasibility problem is presented, since all the constraints considered in the following are of this type.

The presentation of the method is divided into three parts: the first one gives the general principles of the algorithm; the second one brings some details on the operations happening at each step; the third part explains how the parameters of a new hyperplane are computed.

3.1 Algorithm

Consider the following feasibility problem[2]:

$$\text{Find } x \quad \text{subj to} \quad \mathcal{S}_x > 0 \tag{6}$$

[2]The decision variables are denoted x and y in this section: since no state-space representation is involved here, any confusion is avoided.

where x is the vector of decision variables, and $\mathcal{S}_x$ is a real symmetric matrix expressing a set of constraints on matrix form. The problem (6) is convex if $\mathcal{S}_x$ is affine on x. It can be reformulated into an equivalent eigenvalue maximization problem:

$$\sup_{x,y} \; y \quad \text{subj to} \quad \begin{cases} \mathcal{S}_x - yI > 0 \\ \\ y < 1 \end{cases} \tag{7}$$

The problem (7) is feasible if $y > 0$. From (7) a concave function is defined:

$$q(x) := \sup \{y : \mathcal{S}_x - yI > 0, y < 1\} \tag{8}$$

Using $q(x)$, problem (8) can be replaced by the equivalent optimization problem:

$$y_{opt} = \sup_x q(x) \tag{9}$$

In [9], a technique has been presented by Kao, which is suitable for Automatic Control problems, which involves Linear Programming (LP). The function $q(x)$ is bounded iteratively by a set of hyperplanes, leading to a piecewise linear function $p_k(x)$:

$$q(x) \leq p_k(x) := \min_{1 \leq i \leq k} \{a_i^T x + b_i\} \tag{10}$$

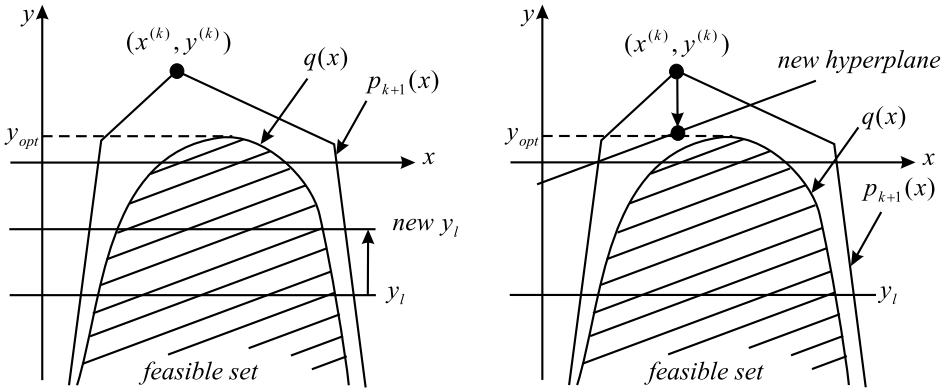
In the following, it is assumed that there exists a mechanism which checks the constraints and generates the hyperplanes (such a mechanism will be derived in the next section). The algorithm begins with an initial value y_l belonging to the feasible set. At iteration k the following LP problem is solved:

$$\max_{x_{\min} \leq x \leq x_{\max}} p_k(x) \tag{11}$$

with $x_{\min}$ and $x_{\max}$ defining some numerical limits on the components of vector x. Let $y^{(k)}$ be the solution of this problem. A linear interpolation involving a parameter $\alpha \in [0, 1]$ derives a new value of y:

$$\hat{y}^{(k)} = \alpha y^{(k)} + (1 - \alpha) y_l \tag{12}$$

If the set of constraints $\mathcal{S}_x - \hat{y}^{(k)} I > 0$ is verified (figure 2(a)), the value of y_l is replaced by $\hat{y}^{(k)}$ else, new hyperplanes are added (figure 2(b)), so that a new LP problem can be solved at iteration $k + 1$. The principle of the CPA is very simple, but the main task is to verify the constraints and to generate the hyperplanes.



(a) (b)

Figure 2: The CPA in the scalar case.

3.2 The mechanism for verifying the constraints and generating the hyperplanes

The verification of the constraints and the generation of the hyperplanes are linked, so that there are considered in the same mechanism.

Two types of constraints have to be considered: in the first case, the constraints are explicit translations of the specifications, so that the verification is done by directly computing the eigenvalues of the corresponding symmetric matrix. A second case arises when the constraints are translated using some equivalent proposition, such as the bounded real lemma for $\mathcal{H}_\infty$ constraints or the Kalman-Yakubovich-Popov (KYP) lemma for passivity constraints: they allow to replace an infinite number of frequency dependent constraints by a unique one. Let us consider the use of the KYP lemma [4]:

Lemma 1 ***(KYP lemma).*** *Let $G(j\omega) = (j\omega I - A)^{-1}B$, where A is a Hurwitz matrix and the pair [A,B] is stabilisable. For any real matrices $Q = Q^T$ and $R = R^T$, the following statements are equivalent:*

1. $H(\omega) = \begin{pmatrix} G(j\omega)^* & I \end{pmatrix} \begin{pmatrix} Q & F \\ F^T & R \end{pmatrix} \begin{pmatrix} G(j\omega) \\ I \end{pmatrix} > 0$ *for all* $\omega \in [0, \infty]$.

2. *$R>0$ and the Hamiltonian* $\mathbf{H} = \begin{pmatrix} A - BR^{-1}F^T & BR^{-1}B^T \\ Q - FR^{-1}F^T & -A^T + FR^{-1}B^T \end{pmatrix}$ *has no eigenvalues on the imaginary axis.*

3. *there exists* $P = P^T$ *such that* $\begin{pmatrix} PA + A^TP & PB \\ B^TP & 0 \end{pmatrix} - \begin{pmatrix} Q & F \\ F^T & R \end{pmatrix} < 0.$

The third proposition is the well-known one. However, the second proposition allows to check the first frequency-dependent constraint by simply computing the eigenvalues of $\mathbf{H}$, without introducing the additional variable P. Furthermore, if the Hamiltonian has some eigenvalues on the imaginary axis, they can be reported in the first proposition as the frequencies where the first constraint is not satisfied.

The generation of the hyperplanes is done using the eigenvectors associated to the negative eigenvalues of the matrix $\mathcal{S}_x - \hat{y}^{(k)}I$. For each negative eigenvalue λ_i an hyperplane is generated from the associated eigenvector v_i, which verifies:

$$v_i^T(\mathcal{S}_x - \hat{y}^{(k)}I)v_i < 0 \tag{13}$$

Since $\mathcal{S}_x$ is affine in x, the quadratic product $v_i^T(\mathcal{S}_x)v_i$ has the form:

$$v_i^T(\mathcal{S}_x)v_i = a_i^T x + b_i \tag{14}$$

and an hyperplane corresponding to the new added constraint is described by:

$$a_i^T x + b_i - \left(v_i^T v_i\right) y > 0 \tag{15}$$

3.3 Determination of the parameters of an hyperplane

To compute the vector a_i and the scalar b_i which define an hyperplane, the matrix $\mathcal{S}_x$ is separated into two matrices:

$$S_x = F(x) + F_0 \tag{16}$$

where all decisions variables x are involved into $F(x)$ and F_0 contains only the additional

terms. Matrices C_{zw} and D_{zw} are similarly rewritten as:

$$\begin{aligned} C_{zw} &= C_0 + C(x) \\ D_{zw} &= D_0 + D(x) \end{aligned} \tag{17}$$

with :

$$\begin{aligned} C_0 &= \begin{pmatrix} C_{11} & 0 & \cdots & 0 \end{pmatrix} \\ D_0 &= D_{11} \\ C(x) &= \begin{pmatrix} C_{Q_{1,1}\otimes T_{1,1}} & \cdots & C_{Q_{m_2,p_2}\otimes T_{m_2,p_2}} \end{pmatrix} \begin{pmatrix} 0 & \\ \vdots & I \\ 0 & \end{pmatrix} = \mathcal{C}N \\ D(x) &= \sum_{j,u} D_{Q_{j,u}} \otimes D_{T_{j,u}} = \mathcal{D} \end{aligned} \tag{18}$$

The vector of decision variables is obtained by bringing together the coefficients of matrices $\mathcal{C}$ and $\mathcal{D}$ as:

$$x^T = \begin{pmatrix} C_{Q_{1,1}} & D_{Q_{1,1}} & \cdots & C_{Q_{m_2,p_2}} & D_{Q_{m_2,p_2}} \end{pmatrix} \tag{19}$$

Combining the definition of a_i, b_i with the decomposition of S_x leads to:

$$v_i^T S_x v_i = v_i^T F(x) v_i + v_i^T F_0 v_i = a_i^T x + b_i \tag{20}$$

The parameter b_i is immediately deduced as:

$$b_i = v_i^T F_0 v_i \tag{21}$$

The rest of this part is dedicated to extract x from $v_i^T F(x) v_i$ to deduce the vector a_i. To this end, it is rewritten as:

$$v_i^T F(x) v_i = \sum_{j=1}^{n_C} v_{iCg}^{(j)T} M_{Cg}^{(j)} \mathcal{C} M_{Cd}^{(j)} v_{iCd}^{(j)} + \sum_{j=1}^{n_D} v_{iDg}^{(j)T} M_{Dg}^{(j)} \mathcal{D} M_{Dd}^{(j)} v_{iDd}^{(j)} \tag{22}$$

where:

- n_C and n_D are respectively the number of appearances of $\mathcal{C}$ and $\mathcal{D}$ on $F(x)$.

- $M_{Cg}^{(j)}$ and $M_{Dg}^{(j)}$ are the matrices multiplying respectively $\mathcal{C}$ and $\mathcal{D}$ on the left in the j-th term
- $M_{Cd}^{(j)}$ and $M_{Dd}^{(j)}$ are similarly the matrices multiplying $\mathcal{C}$ and $\mathcal{D}$ on the right.
- $v_{iCg}^{(j)T}$ and $v_{iDg}^{(j)T}$ point out the parts of $v_i^{(j)T}$ multiplying respectively $\mathcal{C}$ and $\mathcal{D}$ on the left.
- $v_{iCd}^{(j)}$ and $v_{iDd}^{(j)}$ point out the parts of $v_i^{(j)}$ multiplying $\mathcal{C}$ and $\mathcal{D}$ on the right.

Let now introduce the matrices $\hat{C}$ and $\hat{D}$:

$$\begin{aligned}
\hat{C} &= \begin{pmatrix} \begin{pmatrix} C_{Q_{1,1}} \otimes I_{m_1} & 0 \\ 0 & D_{Q_{1,1}} \otimes C_{T_{1,1}} \end{pmatrix} & & 0 \\ & \ddots & \\ 0 & & \begin{pmatrix} C_{Q_{m_2,p_2}} \otimes I_{m_1} & 0 \\ 0 & D_{Q_{m_2,p_2}} \otimes C_{T_{m_2,p_2}} \end{pmatrix} \end{pmatrix} \\
\hat{D} &= \begin{pmatrix} D_{Q_{1,1}} \otimes D_{T_{1,1}} \\ \vdots \\ D_{Q_{m_2,p_2}} \otimes D_{T_{m_2,p_2}} \end{pmatrix}
\end{aligned} \tag{23}$$

Using these definitions, (22) is rewritten as:

$$\begin{aligned}
v_i^T F(x) v_i &= \sum_{j=1}^{n_C} v_{iCg}^{(j)T} \begin{pmatrix} M_{Cg}^{(j)} & \cdots & M_{Cg}^{(j)} \end{pmatrix} \hat{C} \begin{pmatrix} M_{Cd}^{(j)} \\ \vdots \\ M_{Cd}^{(j)} \end{pmatrix} v_{iCd}^{(j)} + \\
&\quad \sum_{j=1}^{n_D} v_{iDg}^{(j)T} \begin{pmatrix} M_{Dg}^{(j)} & \cdots & M_{Dg}^{(j)} \end{pmatrix} \hat{D} M_{Dd}^{(j)} v_{iDd}^{(j)}
\end{aligned} \tag{24}$$

Taking out the vector of the decision variables x leads to:

$$\begin{aligned}
v_i^T F(x) v_i &= x^T \sum_{j=1}^{n_C} \hat{a}_{iC}^{(j)T} \begin{pmatrix} M_{Cd}^{(j)} \\ \vdots \\ M_{Cd}^{(j)} \end{pmatrix} v_{iCd}^{(j)} + x^T \sum_{j=1}^{n_D} \hat{a}_{iD}^{(j)T} M_{Dd}^{(j)} v_{iDd}^{(j)} \\
&= x^T a_{iC} + x^T a_{iD}
\end{aligned} \tag{25}$$

with :

$$\begin{aligned}
\hat{a}_{iC}^{(j)T} &= \begin{pmatrix} \begin{pmatrix} I_{n_q} \otimes (v_{iCg}^{(j)T} M_{Cg}^{(j)}) & 0 \\ 0 & v_{iCg}^{(j)T} M_{Cg}^{(j)} \; C_{T_{1,1}} \end{pmatrix} & & 0 \\ & \ddots & \\ 0 & & \begin{pmatrix} I_{n_q} \otimes (v_{iCg}^{(j)T} M_{Cg}^{(j)}) & 0 \\ 0 & v_{iCg}^{(j)T} M_{Cg}^{(j)} \; C_{T_{m_2,p_2}} \end{pmatrix} \end{pmatrix} \\
\hat{a}_{iD}^{(j)T} &= \begin{pmatrix} \begin{pmatrix} 0 \\ v_{iDg}^{(j)T} M_{Dg}^{(j)} \; D_{T_{1,1}} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} 0 \\ v_{iDg}^{(j)T} M_{Dg}^{(j)} \; D_{T_{m_2,p_2}} \end{pmatrix} \end{pmatrix}
\end{aligned} \tag{26}$$

Finally:

$$a_i = a_{iC} + a_{iD} \tag{27}$$

As a result of the above developments, the construction of a new hyperplane is directly deduced using only the following variables:

$$F_0 \;;\; v_{iCg}^{(j)} \;;\; M_{Cg}^{(j)} \;;\; v_{iCd}^{(j)} \;;\; M_{Cd}^{(j)} \;;\; v_{iDg}^{(j)} \;;\; M_{Dg}^{(j)} \;;\; v_{iDd}^{(j)} \;;\; M_{Dd}^{(j)} \tag{28}$$

The next section shows how different contraints can be translated into a suitable form for applying the CPA. For each of them, variables (28) will be given.

4 Constraints formulation of the design specifications

In this work four types of constraints are considered, which can be gathered into two groups: the first one contains the time domain constraints, namely the time response shaping but also the $\mathcal{H}_2$ norm constraint. The second one contains the frequency-dependent constraints, namely the roll-off or $\mathcal{H}_\infty$ norm contraints, and the stability margins (gain and phase margins for SISO models).

In the following, the developments are presented either in continuous or discrete time. The presented version is the *a priori* most suitable one, but it is also indicated how to translate the results to the other case.

4.1 Time response shaping

To impose a particular shape to a time response, most of the works resort to non convex optimization methods or translate the time domain constraints to the frequency domain. The first approach induces a huge calculation time, whereas in the second one, informations are lost and the constraint becomes harsh in most cases. In this section a time domain constraint is considered using a LMI formulation. Although this formulation is appropriate to discrete-time problems, it can also be used for continuous-time systems by taking a high enough number of values of the time response, spaced by a sample time T which is chosen according to the Shannon condition (see example 2 in section 5).

Given a test input sequence, the aim of time response shaping of discrete time systems can be formulated as follows:

$$\begin{array}{ll} (z_i(nT) - \delta(0))^2 < \tau(0), & n = 0, ..., n_0 \\ (z_i(nT) - \delta(1))^2 < \tau(1), & n = n_0 + 1, ..., n_1 \\ \vdots & \vdots \\ (z_i(nT) - \delta(r))^2 < \tau(r), & n = n_{r-1} + 1, ..., n_r \end{array} \tag{29}$$

where z_i is the i^{th} output; $\delta(j), j = 0, ...r$ is the centre of the allowable interval; $\sqrt{\tau(j)}$ is the maximal tolerated deviation; T is the sample time; r is the number of constraints domains ; n_r is the maximal value of time for which constraints are considered.

Figure 3 shows an example of time response shaping for a unit step response, with $r = 3$.

Each contraint of (29) can be treated separately, so only one constraint for a scalar output is considered in the following. Consider the state-space realization of G_{zw} obtained in section 2.2. If input w is given[3], the value of the output z at each instant n can be found using the algebraic formulation:

$$z(nT) = C_{zw} \left(\sum_{k=1}^{n} A_{zw}^{k-1} B_{zw} w_{n-k} \right) + D_{zw} w_n \tag{30}$$

[3]If w represents an unknown disturbance, a worst case signal should be considered.

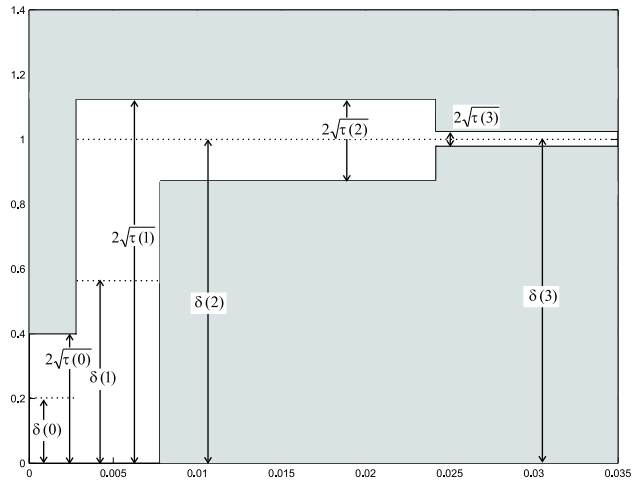


Figure 3: Example of time response constraints

where w_{n-k} is the value of the input at time $n-k$; $z(nT)$ is affine on C_{zw} and D_{zw} (which contains the matrices $C_{Q_{j,u}}$, $D_{Q_{j,u}}$ of the Youla parameter we are looked for). Each constraint of (29) can be written:

$$\left(C_{zw}\left(\sum_{k=1}^{n} A_{zw}^{k-1} B_{zw} w_{n-k}\right) + D_{zw} w_n - \delta(j)\right)^T \left(C_{zw}\left(\sum_{k=1}^{n} A_{zw}^{k-1} B_{zw} w_{n-k}\right) + D_{zw} w_n - \delta(j)\right) < \tau(j) \quad (31)$$

Inequality (31) is not affine in C_{zw} and D_{zw}, but an equivalent LMI formulation is obtained by applying the Schur lemma:

$$\begin{pmatrix} 1 & * \\ \left(C_{zw}\left(\sum_{k=1}^{n} A_{zw}^{k-1} B_{zw} \hat{w}_{n-k}\right) + D_{zw}\hat{w}_n - \hat{\delta}(j)\right) & 1 \end{pmatrix} > 0 \quad (32)$$

with $\hat{w}_{n-k} = \dfrac{w_{n-k}}{\sqrt{\tau(j)}}$, $\hat{w}_n = \dfrac{w_n}{\sqrt{\tau(j)}}$ and $\hat{\delta}(j) = \dfrac{\delta(j)}{\sqrt{\tau(j)}}$.

The constraint (32) is duplicated as much as necessary, according to the number of outputs and the number of samples of each output which are considered in the specifications (29). As an example, for a step input, only constraints corresponding to the transient response and a small part of the permanent response have to be introduced, because the closed-loop plant is guaranteed to be stable.

The verification of the constraint is done directly by computing the eigenvalues of the matrix in (32). Note that since the constraint to be checked in the CPA is actually $\mathcal{S}_x - \hat{y}^{(k)} I > 0$, the first element of matrix (32) has to be replaced by $1 - \hat{y}^{(k)}$.

The new hyperplanes are generated by considering the eigenvectors associated to the negative eigenvalues of (32) (with again the first element of the matrix replaced by $1-\hat{y}^{(k)}$). Only the worst overshoot for each value of j is considered in order to reduce the number of new hyperplanes.

Variables (28) defining a new hyperplane are given by:

$$F_0 = \begin{pmatrix} 1-\hat{y}^{(k)} & * \\ \left(C_0\left(\sum_{k=1}^{n} A_{zw}^{k-1} B_{zw}\hat{w}_{n-k}\right) + D_0\hat{w}_n - \hat{\delta}(j)\right) & 1 \end{pmatrix}$$

C	D
$v_{iCg} = v_{i2}$	$v_{iDg} = v_{i2}$
$M_{Cg} = 2$	$M_{Dg} = 2$
$v_{iCd} = v_{i1}$	$v_{iDd} = v_{i1}$
$M_{Cd} = N\left(\sum_{k=1}^{n} A_{zw}^{k-1} B_{zw}\hat{w}_{n-k}\right)$	$M_{Dd} = \hat{w}_n$

(33)

where v_{i1} and v_{i2} are respectively the first and the second line of v_i, and matrix N is defined in (18).

4.2 $\mathcal{H}_2$ norm constraint

This type of constraint is useful for minimizing the effect of measurement noise on some selected variables, such as the control input for instance. In the following, the discrete time case is developed but the continuous case can be considered with some minor substitutions.

Consider first the case where the output z is scalar. The $\mathcal{H}_2$ norm of the closed-loop plant can be written as:

$$\|G_{zw}\|_2^2 = C_{zw} W_c C_{zw}^T + D_{zw} D_{zw}^T \tag{34}$$

where W_c is the controllability gramian, which is commonly computed as the solution

of the Lyapunov equation[4]:

$$A_{zw} W_c A_{zw}^T - W_c + B_{zw} B_{zw}^T = 0 \tag{35}$$

As it can be noticed, the controllability gramian W_c only depends on matrices A_{zw} and B_{zw} which contain no decision variables, so that W_c is computed only one time.

Using (34), the following equivalence is obtained:

$$\|G_{zw}\|_2^2 < \xi \quad \Leftrightarrow \quad C_{zw} W_c C_{zw}^T + D_{zw} D_{zw}^T - \xi < 0 \tag{36}$$

Writing $W_c = W_c^{1/2} (W_c^{1/2})^T$ and using the Schur lemma, one finally obtains the equivalent constraint:

$$\begin{pmatrix} 1 & * \\ \frac{1}{\sqrt{\xi}} \begin{pmatrix} (W_c^{1/2})^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} C_{zw}^T \\ D_{zw}^T \end{pmatrix} & I \end{pmatrix} > 0 \tag{37}$$

For $dim(z) = m_1$, the $\mathcal{H}_2$ norm can be written as the sum of the norms related to each scalar output:

$$\|G_{zw}\|_2^2 = \sum_{j=1}^{m_1} \|G_{z_j w}\|_2^2 \tag{38}$$

Thus the constraint (37) can be generalized for the multi-outputs case as:

$$\begin{pmatrix} 1 & * \\ \frac{1}{\sqrt{\xi}} \begin{pmatrix} (W_c^{1/2})^T & 0 \\ 0 & I \end{pmatrix} \otimes I_{m_1} \begin{pmatrix} C_{z_1 w}^T \\ \vdots \\ C_{z_{m_1} w}^T \\ D_{z_1 w}^T \\ \vdots \\ D_{z_{m_1} w}^T \end{pmatrix} & I \end{pmatrix} > 0 \tag{39}$$

Contrary to the commonly used LMI formulation which involves a symmetric matrix with the same size of A_{zw}, no additional variable is added in (37) or (39); using these LMIs is thus particularly interesting for large state-space realizations.

[4]It is also well known that for the continuous-time case, D_{zw} has to be equal to 0, while the later equation becomes: $A_{zw} W_c + W_c A_{zw}^T + B_{zw} B_{zw}^T = 0$

As for time response shaping, the verification of the constraint (37) or (39) is done directly by computing the eigenvalues of the matrix, with again the first element replaced by $1-\hat{y}^{(k)}$. The new hyperplanes generated from the eigenvectors associated to the negative eigenvalues are defined by:

$$F_0 = \begin{pmatrix} 1-\hat{y}^{(k)} & * \\ \frac{1}{\sqrt{\xi}} \begin{pmatrix} (W_c^{1/2})^T & 0 \\ 0 & I \end{pmatrix} \otimes I_{m_1} \begin{pmatrix} C_{0z_1}^T \\ \vdots \\ C_{0z_{m_1}}^T \\ D_{0z_1}^T \\ \vdots \\ D_{0z_{m_1}}^T \end{pmatrix} & I \end{pmatrix}$$

where C_{0z_i} and D_{0z_i} $(i = 1, \ldots, m_1)$ are the i-th lines of matrices C_0 et D_0 defined in (18) respectively, and:

C	D
$v_{iCg}^{(j)} = v_{i1}$	$v_{iDg}^{(j)} = v_{i1}$
$M_{Cg}^{(j)} = 2e_j^T$	$M_{Dg}^{(j)} = 2e_j^T$
$v_{iCd}^{(j)} = v_{i2}$	$v_{iDd}^{(j)} = v_{i2}$
$M_{Cd}^{(j)} = \frac{1}{\sqrt{\xi}} W_c^{1/2} N$	$M_{Dd}^{(j)} = \frac{1}{\sqrt{\xi}}$

$j = 1, \ldots, m_1$ (40)

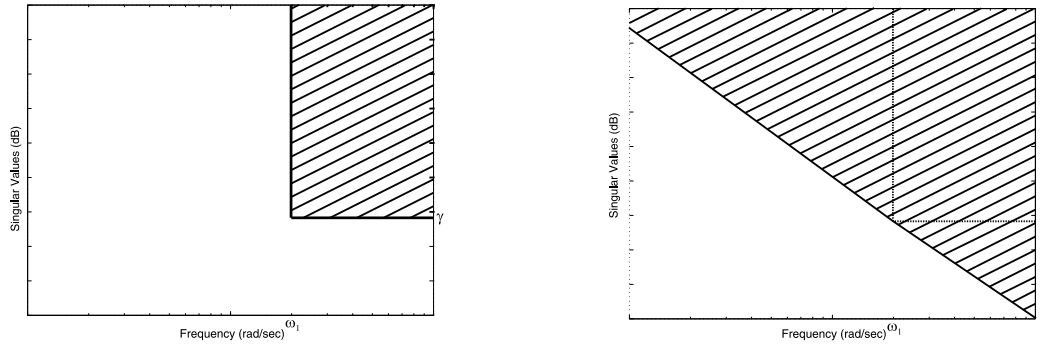
where $v_i^T = \left(v_{i1}, v_{i2}^T\right)$ with v_{i1} scalar, and $e_j \in \mathbb{R}^{m_1}$ is the unitary vector whose j-th element is equal to 1.

4.3 Roll-off shaping and $\mathcal{H}_\infty$ norm constraint

A large number of specifications usually concern the attenuation of some closed-loop transfer function. In $\mathcal{H}_\infty$ synthesis for instance, $\mathcal{H}_\infty$ constraints are introduced on different transfer functions such as the sensitivity function, the complementary sensitivity function,... More generally, it can be interesting to introduce gain constraints in some particular frequency range. Consider for instance the problem of bending mode attenuation: in

most cases, a frequency-dependent weighting filter is applied to a well chosen output of the plant, which induces a distorsion of the initial objective. If for instance the objective is to impose a roll-off constraint at high frequencies (figure 4 (a)), applying some weighting filter will result in the constraint given on figure 4 (b), which induces undesired effects at low frequencies.

The continuous time case is considered in this subsection. The discrete time case can be equivalently handled by applying Tustin transformations.



(a)Exact (b)Relaxed

Figure 4: Roll-off constraint for $\omega \in [\omega_1, \infty)$.

The idea which allows consideration of a particular frequency range is derived from a slight modification of the KYP lemma:

Lemma 2 ***(KYP modified) [8].*** *With the notations and hypotheses given in Lemma 1, the following statements are equivalent:*

1. $H(\omega) > 0$, $\forall \omega \in [\omega_1, \omega_2]$.
2. R *is invertible,* $\exists \omega_i \in [\omega_1, \omega_2]$ *such that* $H(\omega_i) > 0$*, and the Hamiltonian* $\mathbf{H}$ *has no eigenvalues on the imaginary axis belonging in* $[j\omega_1, j\omega_2]$.

Consider now the problem of frequency domain attenuation under gain value γ, that is $\bar{\sigma}(G_{zw})$ has to be less than some specified value γ in some frequency range $[\omega_1, \omega_2]$ (with ω_1 possibly 0 and ω_2 possibly $+\infty$): matrices Q, F and R related to this frequency dependent

constraint are:

$$\begin{aligned} Q &= -C_{zw}^T C_{zw} \\ F &= -C_{zw}^T D_{zw} \\ R &= -D_{zw}^T D_{zw} + \gamma^2 I \end{aligned} \tag{41}$$

From (41), the constraint $H(\omega) > 0$ becomes affine in C_{zw} and D_{zw} by using the Schur lemma:

$$\hat{H}(\omega) = \begin{pmatrix} I & * \\ \frac{1}{\gamma}\begin{pmatrix} B_{zw}^T \left(-j\omega I - A_{zw}^T\right)^{-1} & I \end{pmatrix}\begin{pmatrix} C_{zw}^T \\ D_{zw}^T \end{pmatrix} & I \end{pmatrix} > 0 \tag{42}$$

The specification is checked by applying proposition 2 of Lemma 2. If the Hamiltonian has an eigenvalue $j\hat{\omega}$ belonging in $[j\omega_1, j\omega_2]$, new hyperplanes are generated by injecting the value $j\hat{\omega}$ in (42) and considering the eigenvectors associated with the negative eigenvalues of $\hat{H}(\hat{\omega})$. As for the precedent constraints, the m_1 first elements of (42) have to be replaced by $1 - \hat{y}^{(k)}$.

Variables (28) defining the new hyperplanes are given by:

$$F_0 = \begin{pmatrix} (1-\hat{y}^{(k)})I & * \\ \frac{1}{\gamma}\begin{pmatrix} B_{zw}^T \left(-j\omega I - A_{zw}^T\right)^{-1} & I \end{pmatrix}\begin{pmatrix} C_0^T \\ D_0^T \end{pmatrix} & I \end{pmatrix}$$

C	D
$v_{iCg} = v_{i1}^T$	$v_{iDg} = v_{i1}^T$
$M_{Cg} = \frac{2}{\gamma}$	$M_{Dg} = \frac{2}{\gamma}$
$v_{iCd} = v_{i2}$	$v_{iDd} = v_{i2}$
$M_{Cd} = N\left(j\omega I - A_{zw}\right)^{-1} B_{zw}$	$M_{Dd} = I$

(43)

where $v_i^T = \left(v_{i1}^T, v_{i2}^T\right)$ and $dim(v_{i1}) = m_1$.

4.4 Stability margins

In this subsection both gain and phase margins constraints for SISO plants will be described by LMIs. Continuous time plants will be considered, but the case of discrete time ones can

be equivalently handled by applying Tustin transforms.

Using a suitable LFT form (which will be defined below for each margin) enables to consider that the margin specification is satisfied iff the closed-loop plant remains stable for any scalar uncertainty $\delta \in [0, 1]$. Thus the Nyquist criterion can be used on the plant G_{zw} looped by $-\delta$ (figure 5), from which the stability is guaranteed if and only if the Nyquist diagram of G_{zw} does not cut the half line $(-\infty, -1]$ of the real axis. Indeed G_{zw} being stable, this result guarantees that the stability is preserved for all gain reduction, thus for all $\delta \in [0, 1]$.

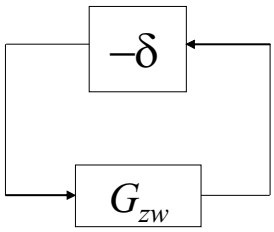


Figure 5: The stability margin formulated as an uncertainty

To derive a convex formulation, the precedent constraint is substituted by a harsher one, where the Nyquist diagram must not go into the half-plane to the left from -1 (figure 6).

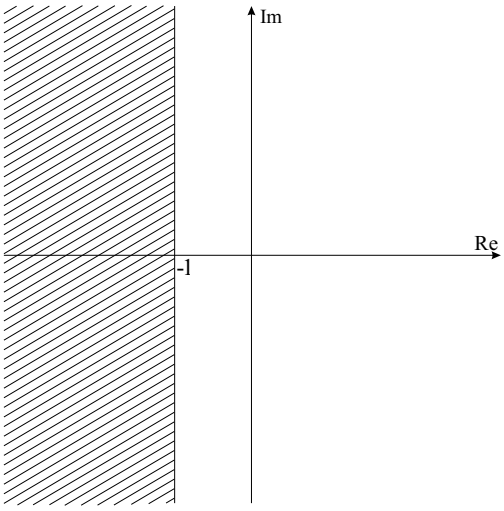


Figure 6: Constraint in the Nyquist plane.

This later constraint directly becomes a passivity condition if G_{zw} is replaced by $G_{zw}+1$:

$$(G_{zw}(j\omega) + 1) + (G^*_{zw}(j\omega) + 1) \geq 0 \qquad \forall \omega \in [0, \infty) \tag{44}$$

Applying lemma 1, the matrices Q, F and R related to this frequency-dependent constraint are:

$$\begin{aligned} Q &= 0 \\ F &= C_{zw}^T \\ R &= D_{zw}^T + D_{zw} + 2 \end{aligned} \tag{45}$$

From (45) the frequency-dependent constraint $H(\omega) > 0$ in lemma 1 is affine in C_{zw} and D_{zw}, and thus in the matrices $C_{Q_{ju}}$ and $D_{Q_{ju}}$ we are looked for. The constraint being checked by computing the eigenvalues of the Hamiltonian, if some of them belongs to the imaginary axis, they are reported in $H(\omega)$ which in that case is scalar. Taking the eigenvector v_i as the scalar 1, variables (28) defining the corresponding hyperplane are given by:

$$F_0 = \begin{pmatrix} B_{zw}^T(-j\omega I - A_{zw}^T)^{-1} & I \end{pmatrix} \begin{pmatrix} 0 & C_0^T \\ C_0 & D_0 + D_0^T + 2 - \hat{y}^{(k)} \end{pmatrix} \begin{pmatrix} (j\omega - A_{zw})^{-1} B_{zw} \\ I \end{pmatrix}$$

C	D
$M_{Cg} = 2$	$M_{Dg} = 2$
$M_{Cd} = N\left(j\omega I - A_{zw}\right)^{-1} B_{zw}$	$M_{Dd} = 1$

(46)

The rest of the section will now explain how to formulate the gain and phase margins on the form given in figure 5.

4.4.1 Gain margin

In order to put the gain margin constraint as shown in figure 5, two types of gain margins should be considered: the first one is the Reduction Gain Margin (RGM), which guarantees the stability for gains less then one. The second is the Increasing Gain Margin (IMG), which concerns gains higher than one.

As is well known, the gain margin specification can be checked in both cases by considering the plant given on figure 7, with $G_{22} = \left(\begin{array}{c|c} A & B_2 \\ \hline C_2 & D_{22} \end{array}\right)$, δ varying in $[0, 1]$ and $g = 1 - 10^{\frac{GM}{20}}$, where GM equals either the RGM or IGM with dB unit.

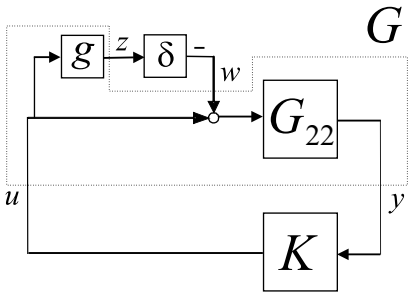


Figure 7: Closed-loop structure for gain margin specification.

The corresponding state space representation of G is then:

$$G: \quad \begin{array}{c} \\ z \\ y \end{array} \begin{array}{c} \begin{array}{cc} w & u \end{array} \\ \left(\begin{array}{c|cc} A & B_2 & B_2 \\ \hline \begin{pmatrix} 0 \dots 0 \end{pmatrix} & 0 & g \\ C_2 & D_{22} & D_{22} \end{array}\right) \end{array} \tag{47}$$

4.4.2 Phase margin

The phase margin is considered by multiplying plant G_{22} by $e^{j\theta}$ (figure 8), where θ is the phase margin.

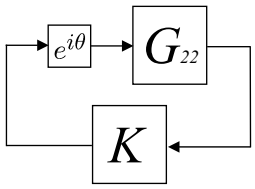


Figure 8: Phase margin representation.

The idea for getting an LFT form is to use a rational parameterization for $e^{j\theta}$:

$$e^{j\theta} = \frac{1 + j\hat{\theta}}{1 - j\hat{\theta}} \tag{48}$$

Note that for $\theta \in [0, \theta_e]$, $\hat{\theta}$ is real and belongs to $\left[0, \frac{sin(\theta_e)}{1+cos(\theta_e)}\right]$.

Equation (48) can be realized as the interconnection $j\hat{\theta} * \mathcal{N}$, with:

$$\mathcal{N} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} \tag{49}$$

The phase margin specification can then be checked by considering the plant given on figure 9, with $\delta \in [0, 1]$.

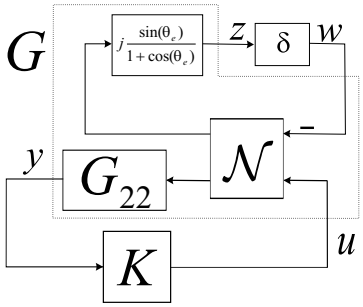


Figure 9: Closed-loop structure for phase margin specification.

The corresponding state space representation of G is then:

$$\begin{array}{c} \\ z \\ y \end{array} \begin{array}{c} \begin{array}{ccc} & w & u \end{array} \\ \left(\begin{array}{c|cc} A & 2B_2 & B_2 \\ \hline \begin{pmatrix} 0 \dots 0 \end{pmatrix} & \hat{g} & \hat{g} \\ C_2 & 2D_{22} & D_{22} \end{array}\right) \end{array} \tag{50}$$

where $\hat{g} = -j\frac{sin(\theta_e)}{1+cos(\theta_e)}$.

5 Numerical validation

To show the interest of the proposed approach, two examples are presented. The first one illustrates the simplicity of applying the methodology developed in this paper by considering $\mathcal{H}_2$ norm constraints together with roll-off specifications for a flexible plant. The second example shows how the time response shaping can be used in the continuous-time case, while introducing a phase margin requirement.

Two parameters have to be defined to use the CPA: initial value y_l has been taken as a sufficient negative value (which will be given for each example), while parameter α involved in relaxation (12) is randomly chosen between 0 and 1 at each iteration.

5.1 Example 1: $\mathcal{H}_2$ control with roll-off constraints

For the first example, let consider the digital control of a hard-disk read/write head system. It is taken from a Matlab demo [7]. The head-disk assembly (HDA) and actuators are modelled by a SISO system where the input is the current i_c driving the voice coil motor and the output is the position error signal $\epsilon_\theta = \theta_{ref} - \theta$. The order of the state space representation is 10 including two rigid-body modes and the first four resonances. The model also includes a small delay $T_r = 10^{-5}$ sec.

Only the rigid modes are considered for the compensator design, although the bending modes are to be attenuated while rejecting an input disturbance. The rigid model is first discretized using a zero-order hold with sample time $T = 7.10^{-5}$ sec. To handle disturbance rejection, a $\mathcal{H}_2$ performance is considered between the input disturbance and the error ϵ_θ. To reduce the control effort, a second $\mathcal{H}_2$ performance is considered between the input disturbance and the control input i_c.

The design follows two steps: the first one considers only the time domain performance of the rigid model through $\mathcal{H}_2$ norm constraints on both channels given above [1]. The second step adds a roll-off constraint to attenuate the bending modes.

The Youla parameter is taken as an FIR filter. Initial value y_l in the CPA is -10^2. The initial compensator is taken from the Matlab demo:

$$K_{init}(z) = \frac{46.29z^2 - 89.32z + 43.09}{z^2 - 0.2801z - 0.7199} \tag{51}$$

Table 1 outlines a comparison of the computation times involved in the first step (pure $\mathcal{H}_2$-synthesis), using a Pentium4 2.53 GHz, for different orders of the Youla parameter: in the first column, the classical $\mathcal{H}_2$-norm formulation (see e.g. [8]) is used together with the Matlab SDP solver [6]; in the second column, the formulation proposed in this paper is used, again with the Matlab SDP solver; the third column corresponds to the proposed formulation solved with the CPA; for this later case, the number of iterations and hyperplanes are given in columns 4 and 5.

Note also that in the first approach, a symmetric matrix of dimension $12 + n_q$ is looked

for, whereas the proposed formulation involves only $n_q + 2$ decision variables.

n_q	classical formulation (Matlab SDP solver)	proposed formulation (Matlab SDP solver)	proposed formulation (CPA solver)		
	Times (sec)			iterations	hyperplanes
1	26.07	0.16	1.23	29	18
2	26.74	0.18	1.26	31	24
3	47.73	0.22	1.31	35	28
4	59.99	0.27	1.34	37	30
5	107.63	0.29	1.38	41	35

Table 1: Comparison of $\mathcal{H}_2$ formulations

Clearly the computation time is much higher when using the classical $\mathcal{H}_2$ formulation. Furthermore the gap increases when the order of the Youla parameter rises.

The SDP solver is faster than the CPA when the same number of decision variables is involved: it is obvious that for a pure $\mathcal{H}_2$ synthesis where the controllability gramian can be computed independently, the CPA offers no advantage. Nevertheless using the CPA allows to prevent introducing additional variables in other specifications like $\mathcal{H}_\infty$ constraints or gain and phase margins: this fact will be illustrated by the second step of the design (see also [10]).

Let now examine the practical results of this first design: with a Youla parameter of order 1, an instability occurs when the resulting controller is applied to the complete model (see figure 10). This instability is detected by the resonance peak which appears on the closed-loop transfer function of the flexible model (figure 11, solid line).

To reduce the resonance peak while preserving the time-domain performance, a gain constraint on the rigid model is considered by minimizing γ in the frequency range $[10^4\ , 3.5.10^4]$, while maintaining the $\mathcal{H}_2$ constraint. The value of y_l in the CPA is taken as -10^4.

Table 2 outlines a comparison of the computation times between a LMI formulation

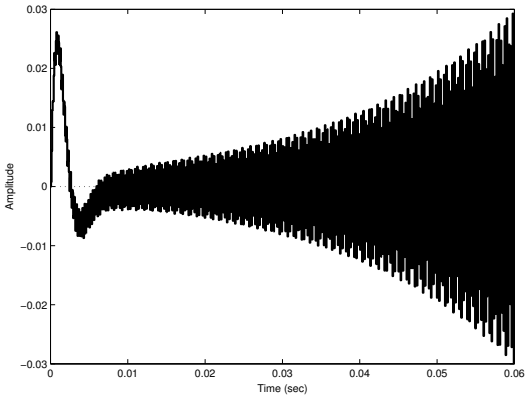


Figure 10: Impulse response of the flexible model with $\mathcal{H}_2$ compensator.

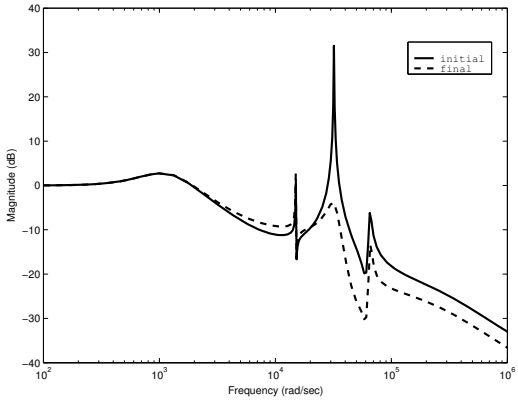


Figure 11: Reduction of the resonance peak (flexible model)

developed in [1] to handle gain constraints with SDP solvers and the proposed formulation (42). In the first case, the Matlab [6] and SeDuMi [5] [14] solvers are successively used; in the second case the number of iterations and hyperplanes are also given.

If the computation time involved in the first case is clearly dependent on the quality of the SDP solver, the proposed formulation is very little expensive; the reason does not come from the efficiency of the solver but from the number of decision variables, which is $2 \times (13 + n_q) \times (14 + n_q)$ in the first case and only $n_q + 2$ in the second. This fact will be emphasized again in the next example.

[5]Self-Dual-Minimisation package developed by Jos F. Sturm. The interface used is the one developed in the LAAS-CNRS.

n_q	formulation with SDP solvers		formulation with CPA solver		
	Matlab	SeDuMi			
	Time (sec)			iterations	hyperplanes
1	194.59	80.33	12.21	426	203
2	321.42	97.34	12.78	421	228

Table 2: Comparison of gain constraints formulations.

In this second design, $n_q = 2$ is necessary to obtain $\gamma = 0.1$ on the rigid model: a significant attenuation is then obtained in the frequency range of the first two flexible modes (figure 12), which is sufficient to reduce the peak resonance of the flexible model (figure 11, dashed line). The time-domain performance is preserved, as shown in figure 13 (dotted line).

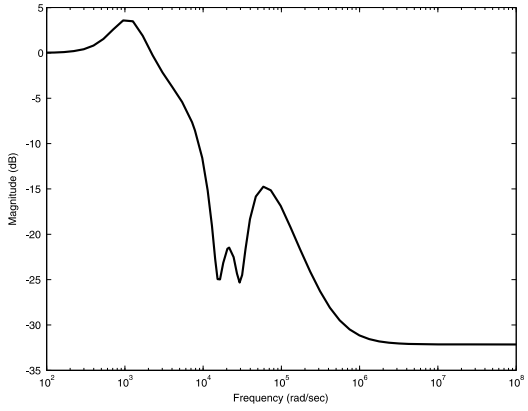


Figure 12: Closed-loop magnitude: rigid model with final controller.

The proposed approach allows therefore to reach the goals with a Youla parameter of order 2 (that is a final controller of order 6).

The improvement brought by using the proposed approach is not considerable, but remains significant. This example especially underlines the simplicity of the synthesis induced by the proposed formulation.

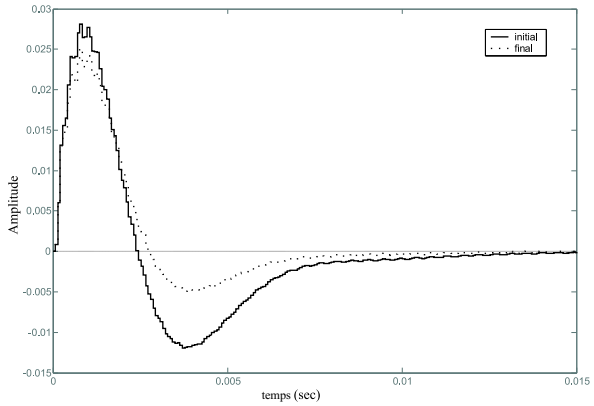


Figure 13: Impulse response of the flexible model with CPA compensator.

5.2 Example 2: time response shaping with phase margin

The second example concerns a simplified overhead travelling crane system. The assembly is modelled by a SIMO system where the input is the tension e driving the DC motor and the outputs are the position x and the angle ϕ around the vertical. It is described by equations (52) below: the order of the state space representation is 5 including three rigid modes and one resonance mode. In the synthesis only two of the rigid modes are considered, because the electrical time-constant L/R of the DC motor can be neglected.

$$\begin{aligned} \frac{dx}{dt} &= \frac{r}{N}\omega \\ J\frac{d\omega}{dt} &= \Phi i + f\omega \\ L\frac{di}{dt} + Ri &= Ae - \Phi\omega \\ l\frac{d^2\phi}{dt^2} + a\frac{d\phi}{dt} + g\phi &= -\frac{d^2x}{dt^2} \end{aligned} \tag{52}$$

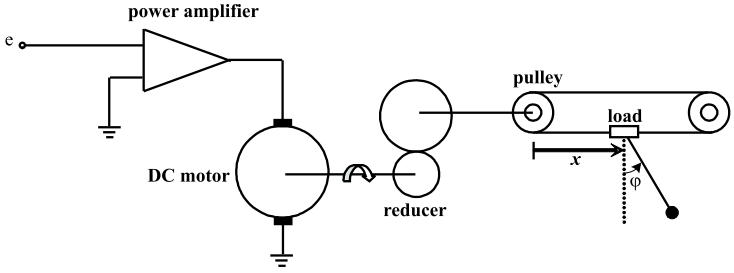


Figure 14: Overhead travelling crane system

The data of the overhead travelling crane system are given in table 3.

Constant name	Symbol	Value
Amplifier gain	A	1
Rotor inductance	L	0.2 mH
Total resistance	R	2.74 ohm
Torque constant	Φ	16.2 mNm/A
Total inertia on motor axis	J	3.06 10^{-6} kgm^2
Friction coefficient on motor axis	f	3.2 10^{-5} Nms
Reduction ratio	N	17
Pulley radius	r	22 mm
Bar length	l	269 mm
Pendulum damping coefficient	a	0.26 m/s
Gravity acceleration	g	9.81 m/s^2

Table 3: Data of the travelling crane system.

The challenge imposed in the manufacturer specifications is to move the load from an extremity to another (0.4 m) with no overshoot in only 1.2 s, with position at least equal to 0.395 m after this date. The control value must not overtake ± 10 V, and the oscillation of the pendulum must not exceed ± 0.25 rad. As stability margin, the phase margin of the system should be more than 35°.

The initial compensator is taken as a static one:

$$K_{init} = (-1 \quad 1) \tag{53}$$

To take into account the time domain specifications, a sample time $T = 2.10^{-3}$ is chosen on interval $[0, 5]$ sec. Note that the number of constraints introduced is equal to 2500 for each of the 3 outputs. For the known SDP solvers, one has then to consider 7500 different LMIs, where only their translation into the software code represents a high computation time. On the contrary, when using the CPA, the time constraints are directly checked

using their natural form (29) and only one of them is used at each iteration to define a new hyperplane.

To define the Youla parameter, the orthogonal basis of the following transfer functions [2] has been considered:

$$Q_i(s) = \frac{\sqrt{2Re(a_i)}}{s + a_i} \prod_{k=1}^{i-1} \frac{s - \bar{a}_k}{s + a_k} \tag{54}$$

The coefficients a_i, which define the poles of the Youla parameter, have to be chosen in accordance with the expected dynamics of the closed-loop plant. To this end, random numbers are selected between 0 and 50, and the order of the Youla parameter is gradually increased until the specifications are met.

In order to examine the sensitivity to this random choice, three different sets corresponding to a Youla parameter of order 10 have been tested (see table 4). The two first dynamics lead to controllers achieving the specifications, although the Bode diagrams of the open-loop are somewhat different (figure 15): the phase margins obtained are 37.1° and 39.9° respectively; in both cases, the output x is brought into the template (figure 16.a) while satisfying the limitations on the control input (figure 16.b) and the angle (figure 16.c).

The third dynamics lead to an infeasible problem. Thus, the choice of the dynamics of the Youla parameter plays a major role on the feasibility decision for a given order of the controller. This is why it is important to use LMI formulations which are less sensitive to the order to be able to test quickly different random choices.

The first dynamics are considered now for time computation comparison. When the CPA is used, the result is obtained with 19.76 sec computation time (using the same processor as in example 1), by means of 155 iterations and 44 hyperplanes. The resulting controller order is 15. The computation time induced by other solvers is significantly more important (1425.9 sec for Matlab solver and 811.47 for SeDuMi) due to two reasons: the first one is the high number of LMI constraints introduced for the time response shaping; the second is the passivity constraint which has to be used to translate the phase margin requirement, which introduces 820 supplementary decision variables.

a_i^1	1.1739	21.093	12.106	43.677	46.896	34.259	23.511	21.158	28.63	21.283
a_i^2	9.5544	38.293	23.706	12.859	16.261	4.2235	33.092	25.825	10.632	24.25
a_i^3	38.6676	11.9158	41.0752	48.4570	42.4739	31.4724	10.3431	36.0532	27.0490	19.6218

Table 4: Dynamics chosen for the Youla parameter

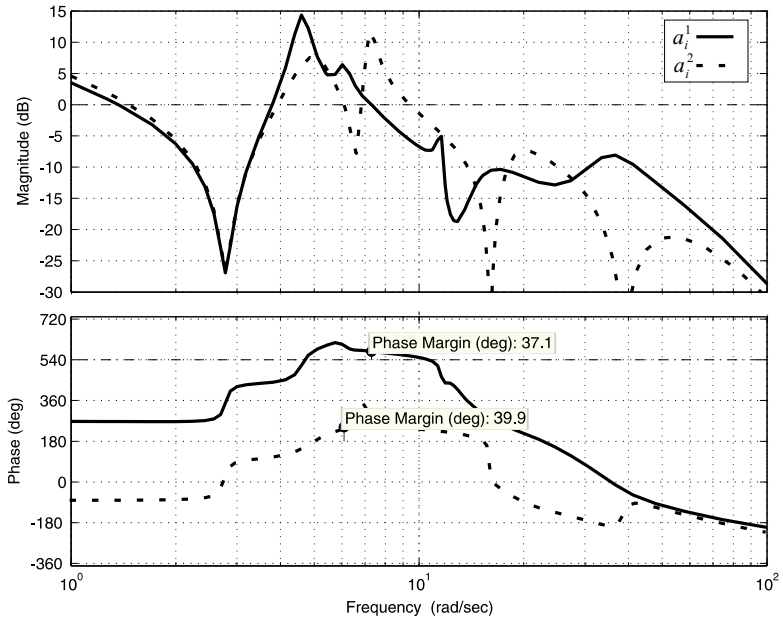
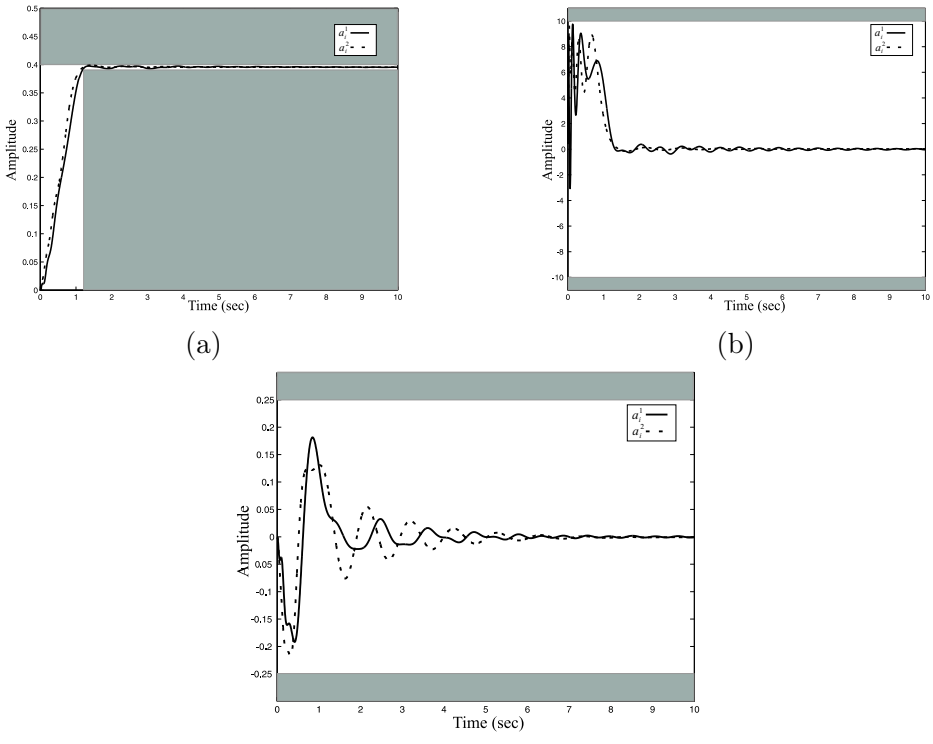


Figure 15: *Frequency response of the open-loop plant.*

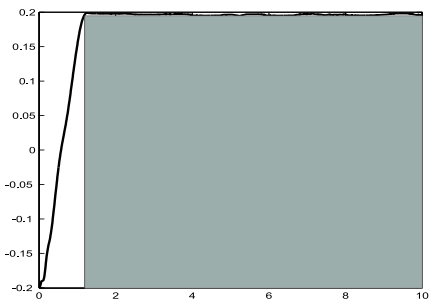


(a) (b)

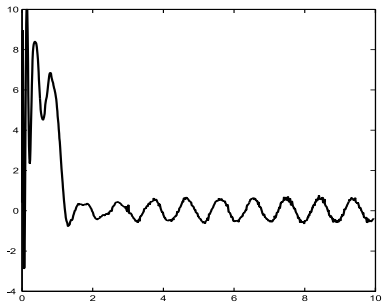
(c)

Figure 16: *Time responses of the simulation model.*

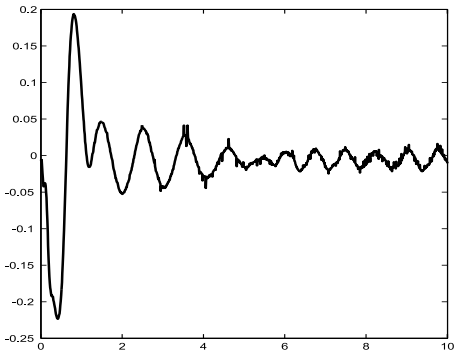
To validate the simulations results, the first controller has been implemented on the real plant. The figures below show the results obtained using the Real Time Toolbox of Matlab [17] by means of a PCI-6024E board. The sample time is equal to 0.01 sec, and the displacement is done from -0.2m to 0.2m, which corresponds to 0.4 m (figure 17). As it can be noticed the results are very close to those given by the simulation model (figure 16), although a limit cycle is observed due to dry friction on the motor axis. Nevertheless the time templates are satisfied.



(a)



(b)



(c)

Figure 17: *Time responses of the real plant.*

6 Conclusion

Designing a controller to meet different specifications or deciding the unfeasibility of the problem can be expressed using the Youla parameterization as a convex feasibility problem: the particular LMI formulations of the constraints brought in this paper are directly

expressed in terms of the decision variables which define the Youla parameter, without using any transformation or relaxation.

The application of the CPA allows to prevent the introduction of additional variables, which implies that a Youla parameter with high order can be considered without numerical difficulties. The simplicity of utilizing the CPA makes it attractive, although some numerical improvements can be a subject of forthcoming works.

The stability margins constraints have been considered in this paper for SISO plants, the extension to the MIMO case being under investigation. Finally, one can note that using the Youla parameterization leads generally to high order controllers: it allows to conclude to the feasibility or non-feasibility of the control problem, but from a practical point of view, controllers of lower order should be preferred. For this reason, developing a suitable reduction method to approximate the Youla parameter by a rational transfer function while still satisfying the constraints will be the subject of forthcoming studies.

References

[1] M. Abbas-turki, G. Duc, and B. Clement. $\mathcal{H}_2/\mathcal{H}_\infty$ control without Lyapunov matrix or frequency weights. *16th IFAC World Congress, Prague*, July 2005.

[2] H. Akcay and B. Ninness. Orthonormal basis functions for continuous-time systems and lp convergence. *Math. Contr. Sign. Syst.*, 12(3):295–305, 1999.

[3] S. Boyd and C. Barratt. *Linear Controller Design - Limits of Performance*. Prentice-hall edition, 1991.

[4] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. Siam edition, 1994.

[5] B. Clement, G. Duc, and S. Mauffrey. Aerospace launch vehicle control: a gain-scheduling approach. *Control Engineering Practice*, 13(3):333–347, March 2005.

[6] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali. *LMI Control Toolbox*. The math works inc. edition, 1995.

[7] A. Grace, A. J. Laub, J. N. Little, and C. M. Thompson. *Control System Toolbox*. The math works inc. edition, 1995.

[8] H. A. Hindi, B. Hassibi, and S. P. Boyd. Multiobjective $\mathcal{H}_2/\mathcal{H}_\infty$-optimal control via finite dimensional Q-parametrization and linear matrix inequality. *IEEE Proceeding of the American Control Conference Philadelphia, Pensylvania*, pages 3244–3249, June 1998.

[9] C.-Y. Kao. *Efficient Computational Methods for Robustness Analysis*. PhD thesis, Massachusetts Institute of Technology, September 2002.

[10] C.-Y. Kao and U. T. Jönsson. An algorithm for solving optimization problems involving special frequency dependent lmis. *American Contr. Conf.*, pages 307–311, June 2000.

[11] J. H. Lee. Nonlinear programming approch to biaffine matrix inequality problems in multiobjective and structured control. *IEEE Proceeding of the American Control Conference San Diego, California*, pages 1817–1821, June 1999.

[12] J. M. Maciejowski. *Multivariable feedback design*. Addison-wesley edition, 1990.

[13] M. C. Oliveira, J. C. Geromel, and J. Bernussou. Extended H_2 and H_∞ characterizations and controller parametrizations for discrete-time systems. *International Journal of Control*, 75(9):666–679, June 2002.

[14] D. Peaucelle, D. Henrion, Y. Labit, and K. Taitz. *SeDuMiInterface 1.04*. Available at http://www.laas.fr/ peaucell/software/SeDuMiInt.html, 2002.

[15] C. Scherer, P. Gahinet, and M. Chilali. Multiobjective output-feedback control via LMI optimization. *IEEE Trans. Automat. Contr.*, 42(7):896–911, 1997.

[16] C. W. Scherer. From mixed to multi-objective control. *Proceedings of the 38th Conference on Decision & Control Phoenix, Arizona USA*, pages 3621–3626, December 1999.

[17] Real-Time Windows Target. The math works inc. edition, 1999.

[18] O. Voinot, D. Alazard, P. Apkarian, S. Mauffrey, and B. Clement. Launcher attitude control: Discrete-time robust design and gain-scheduling. *Control Engineering Practice*, 11(11):1243–1252, November 2003.

[19] E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer, communications and control engineering series edition, 1997.