Normalizing compensators for general H standard problems

Ramine Nikoukhah, François Delebecque

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

1992

25ème anniversaire

N° 1803

Programme 5
Traitement du Signal,
Automatique et Productique

NORMALIZING COMPENSATORS FOR GENERAL $H\infty$ STANDARD PROBLEMS

Ramine NIKOUKHAH
François DELEBECQUE

Décembre 1992

RR-1803

Normalizing compensators for general H_∞ standard problems

Ramine Nikoukhah and François Delebecque

INRIA, Rocquencourt BP 105, 78153 Le Chesnay Cedex, France

Abstract In practical applications of H_∞ control theory, one often encounters standard problems that do not satisfy the assumptions on their pole-zero structure at infinity required for the application of Riccati-based solution methods. In this paper, we present a method based on the idea of pre- and post-compensation for transforming these H_∞ standard problems into problems that do satisfy these assumptions. These transformations are very important, because, solutions to compensated standard problems can be used to construct solutions to the original standard problems very easily. The method presented is very systematic and can easily be implemented.

Compensateurs normalisants pour le problème H_∞ général

Résumé Dans les applications pratiques de la commande H_∞, on rencontre souvent des problèmes standards pour lesquelles les hypothèses sur la structure pôles/zéros à l'infini, qui sont nécessaires à l'application des méthodes fondées sur les équations de Riccati, ne sont pas satisfaites. On présente ici une méthode de pré- et post- compensation qui permet de transformer ces problèmes standards de façon à satisfaire les hypothèses habituelles. La solution au problème original s'obtient alors facilement à partir de la solution du problème compensé. La méthode proposée est systématique et conduit à un algorithme facilement implémentable.

1 Introduction

In this paper, we consider the H_∞ standard problem for the following plant:

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix}$$

where $P_{12}(s)$, $P_{21}(s)$ have respectively full normal column and row ranks, but, they may have poles and zeros at infinity. Moreover, we allow $P_{22}(s)$ to have poles at infinity. As usual, the problem consists in finding, for a given γ, an internally stabilizing controller $K(s)$ such that $\|F_l(P(s), K(s)\|_\infty < \gamma$, where F_l denotes the lower linear fractional transformation. This problem can be solved using Kwakernaak's polynomial approach [7]; here we develop a procedure for solving them using Riccati-based solution method [2, 4].

The main novelty of the solution method presented in this paper is that it can handle improper plants; a number of solutions for singular H_∞ control problem with proper plants (i.e., when $P_{12}(s)$ and $P_{21}(s)$ have infinite zeros but not poles) can be found in the literature [11, 1].

A common situation where H_∞ standard problems with improper plants come up in practice is in the mixed-sensitivity formulation where requirements on the high frequency roll-off of the open-loop gain calls for the use of improper weighting matrices. The usual approach to overcome this problem, is to perturb design parameters in order to obtain a proper plant by introducing "near infinity" poles into the system. This method however has many drawbacks: numerical problems (even though the perturbed problem meets the required assumptions, it is numerically ill-posed), construction not systematic (perturbations must be small enough compared to the modes of the system yet large enough to avoid, as much as possible, numerical ill-posedness) and loss of optimality with no a priori measure of discrepancy.

In our approach, we start by modifying $P(s)$ in such a way as to obtain a standard problem to which usual Riccati-based solutions can be applied (by removing offending poles and zeros at infinity), and, construct its solution. Then, we incorporate the corresponding modifications into the controller and construct a solution to the original problem. This approach is illustrated in Figure 1.

More specifically, the procedure is the following:

1- let

$$\hat{P}(s) = \begin{pmatrix} I & 0 \\ 0 & L(s) \end{pmatrix} P \begin{pmatrix} I & 0 \\ 0 & R(s) \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & -T(s) \end{pmatrix},$$

2- construct a solution $\hat{K}(s)$ to $\hat{P}(s)$, using Riccati-based solution method,

3- finally, let

$$K(s) = R(s)(I + \hat{K}(s)T(s))^{-1}\hat{K}(s)L(s).$$

A similar procedure has been used in [1] for removing zeros on the $j\omega$-axis and at infinity.

Pre- and post-compensators $R(s)$ and $L(s)$ are designed to remove infinite poles and zeros of $P_{21}(s)$ and $P_{12}(s)$ respectively and, $T(s)$ removes the polynomial part (if it exists) of the (2,2)-block of the compensated plant. We say that the standard plant $\hat{P}(s)$ is *normalized* if $\hat{P}_{12}(s)$ and $\hat{P}_{21}(s)$ have no poles and zeros at infinity and $\hat{P}_{22}(s)$ has no poles at infinity. In Section 2, we derive conditions under which this procedure can be used and we show how it can be applied to solve a few simple examples.

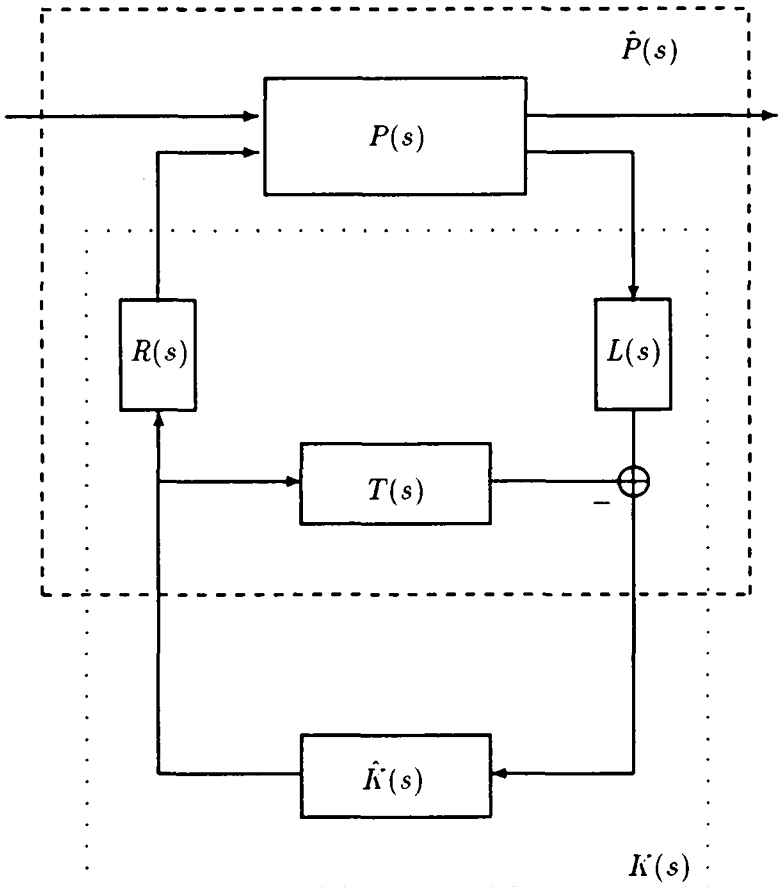

Figure 1

Except in simple cases, the design of these pre- and post-compensators requires a state-space formulation of improper rational matrices. For this, we shall use polynomial state-space descriptions (PSSD): $\Sigma = (A, B, C, D(s))$ is a PSSD representing the transfer function $\Sigma(s)$ if

$$\Sigma(s) = C(sI - A)^{-1}B + D(s),$$

where $D(s)$ is a polynomial matrix. The PSSD is a very natural representation for improper transfer functions which has some advantages over the descriptor system representation [9], at least, for the applications considered in this paper. In Section 3, we review some results on PSSD's that will be needed in the development of the normalization procedure of Section 4.

Finally, in Section 4, we present the normalization procedure, i.e. an algorithm for the construction of pre- and post-compensators. There, we assume that the standard plant $P(s)$ is available in PSSD:

$$P = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12}(s) \\ C_2 & D_{21}(s) & D_{22}(s) \end{array}\right].$$

P may be constructed directly from $P(s)$ (the realization algorithms for the construction of PSSD's from their transfer functions are widely available [5]) or from smaller subsystems using basic PSSD operations [8].

2 Preliminaries

In this section, we present conditions that the pre- and post-compensators must satisfy in order for the normalization procedure presented above to be valid. The following is the usual definition of properness and internal stability [3].

Definition 2.1 *1- We say that $K(s)$ internally properizes $P_{22}(s)$ if, in Figure 2, the four transfer functions from v_1, v_2 to u, y are proper.*

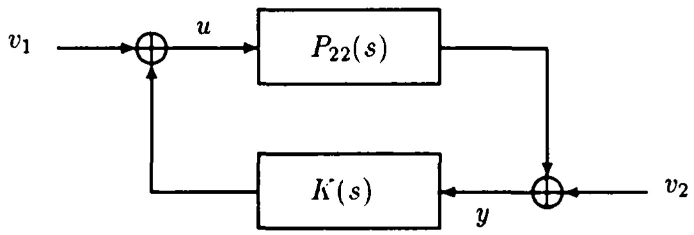

Figure 2

2- We say that $K(s)$ internally stabilizes $P_{22}(s)$ if, in Figure 2, the four transfer functions from v_1, v_2 to u, y are stable.

The following results specifies under what conditions we can apply the normalization procedure. In particular, when can we guarantee that the procedure finds an internally properizing and stabilizing controller that meets the H_∞ norm specification: $||F_l(P(s), K(s))||_\infty < \gamma$ when such a controller does exist.

Theorem 2.1 *Consider the standard problem in Figure 1. Let $L(s)$ and $R(s)$ be square invertible biproper and bistable, and let $T(s) = 0$. Then* *a- if $\hat{K}(s)$ internally properizes and stabilizes $\hat{P}_{22}(s)$ then*

$$K(s) = R(s)\hat{K}(s)L(s) \tag{2.1}$$

internally properizes and stabilizes $P_{22}(s)$,

b- if $K(s)$ internally properizes and stabilizes $P_{22}(s)$ and

$$||F_l(P(s), K(s))||_\infty < \gamma \tag{2.2}$$

then there exists $\hat{K}(s)$ internally properizing and stabilizing $\hat{P}_{22}(s)$ and

$$||F_l(\hat{P}(s), \hat{K}(s))||_\infty < \gamma. \tag{2.3}$$

Proof The proof of part a- is straightforward; in fact, its converse is also true. For part b-, let

$$\hat{K}(s) = R^{-1}(s)K(s)L^{-1}(s). \tag{2.4}$$

Thanks to

$$F_l(P(s), K(s)) = F_l(\hat{P}(s), \hat{K}(s))$$

and the fact that $\hat{K}(s)$ internally properizes and stabilizes $\hat{P}_{22}(s)$, we can immediately deduce b-. □

Corollary 2.1 *In Theorem 2.1, a- and b- hold even if we allow $L(s)$ and $R(s)$ to have poles and zeros on the $j\omega$-axis or at infinity, provided no pole-zero cancellations on the $j\omega$-axis or at infinity occur in the product $L(s)P_{22}(s)R(s)$.*[1]

Proof Part a- can be shown by examining the corresponding transfer functions. The proof is straightforward and is omitted here.

For part b-, the proof is based on a perturbation argument. Let $\hat{K}(s)$ be as in (2.4), if no pole-zero cancellations occur on the $j\omega$-axis or at infinity in $R(s)\hat{K}(s)L(s)$, $\hat{K}(s)$ internally properizes and stabilizes $\hat{P}_{22}(s)$ and (2.3) is satisfied. If there are pole-zero cancellations on the $j\omega$-axis or at infinity, these cancelled modes (poles and zeros) appear in closed-loop transfer functions and thus violate internal stability or properness. In that case, perturb $\hat{K}(s)$ as follows:

$$\hat{K}(s) \leftarrow \hat{K}\left(\frac{s+\epsilon}{1+\epsilon s}\right).$$

It can be verified that for sufficiently small ϵ, this perturbed $\hat{K}(s)$ internally properizes and stabilizes $\hat{P}_{22}(s)$ and satisfies (2.3). □

Conditions in Corollary 2.1 are only sufficient conditions, i.e., even if pole-zero cancellations do occur in $L(s)P_{22}(s)R(s)$, $K(s)$ in a- may properize and stabilize $P_{22}(s)$. This happens if exactly the same cancellations occur in $R(s)\hat{K}(s)L(s)$.

[1] In fact, part a- holds even if $L(s)$ and $R(s)$ are allowed to have unstable poles and zeros provided no unstable pole-zero cancellations occur in $L(s)P_{22}(s)R(s)$.

Example 1 Let

$$P_{22}(s) = \frac{1}{s+1}, \quad R(s) = s+1 \text{ and } L(s) = 1.$$

Then $\hat{K}(s) = -1$ internally properizes and stabilizes $\hat{P}_{22}(s) = 1$. However

$$K(s) = R(s)\hat{K}(s)L(s) = -(s+1)$$

does not internally properize $P_{22}(s)$ because

$$\frac{K(s)}{1 - P_{22}(s)K(s)} = -\frac{s+1}{2}$$

is not proper. On the other hand,

$$\hat{K}(s) = -1/s$$

also internally stabilizes $\hat{P}_{22}(s)$ and

$$K(s) = -\frac{s+1}{s}$$

internally properizes and stabilizes $P_{22}(s)$. That is because the pole-zero cancellation at infinity occurring in $P_{22}(s)R(s)$ also occurs in $R(s)\hat{K}(s)$.

Corollary 2.2 *In Theorem 2.1, if $T(s)$ is a polynomial matrix and $L(s)$ and $R(s)$ are allowed to have poles and zeros at infinity. Then*

a- if $\hat{K}(s)$ internally stabilizes $\hat{P}_{22}(s)$ then

$$K(s) = R(s)(I + \hat{K}(s)T(s))^{-1}\hat{K}(s)L(s) \tag{2.5}$$

(if it exists) internally stabilizes $P_{22}(s)$,

b- if $K(s)$ internally stabilizes $P_{22}(s)$ and

$$||F_l(P(s), K(s))||_\infty < \gamma \tag{2.6}$$

then there exists $\hat{K}(s)$ internally stabilizing $\hat{P}_{22}(s)$ and

$$||F_l(\hat{P}(s), \hat{K}(s))||_\infty < \gamma. \tag{2.7}$$

Proof Part a- is straightforward. For part b-, let

$$\hat{K}(s) = R^{-1}(s)K(s)L^{-1}(s)(I - T(s)R^{-1}(s)K(s)L^{-1}(s))^{-1}. \tag{2.8}$$

Then, $\hat{K}(s)$ internally stabilizes $\hat{P}_{22}(s)$ and satisfies (2.7). □

Now that we know conditions under which our normalization method can be used, let us consider a few simple examples for which pre- and post-compensators can be "guessed". A systematic normalization algorithm will be presented in Section 4.

Example 2 Consider the H_∞ control problem with improper plant [7]:

$$P(s) = \left(\begin{array}{c|c} \frac{1+\sqrt{2}s+s^2}{s^2} & \frac{1}{s^2} \\ 0 & 0.1+0.1s \\ \hline \frac{1+\sqrt{2}s+s^2}{s^2} & \frac{1}{s^2} \end{array}\right).$$

The infinite pole of $P_{12}(s)$ can be removed by letting $R(s) = 1/(s+2)$: the resulting normalized problem is:

$$\hat{P}(s) = \left(\begin{array}{c|c} \frac{1+\sqrt{2}s+s^2}{s^2} & \frac{1}{2s^2+s^3} \\ 0 & \frac{0.1+0.1s}{2+s} \\ \hline \frac{1+\sqrt{2}s+s^2}{s^2} & \frac{1}{2s^2+s^3} \end{array}\right)$$

for which we can construct the optimal central controller

$$\hat{K}(s) = (s+2)\frac{-8.0997 - 15.634s}{15.010 + 5.5936s + s^2}.$$

$K(s)$ is then obtained as follows:

$$K(s) = \frac{1}{s+2}\hat{K}(s) = \frac{-8.0997 - 15.634s}{15.010 + 5.5936s + s^2},$$

which is the same result as the one found in [7].

In general, if we require internal properness, we are not allowed to use improper $L(s)$ or $R(s)$ since this would either lead to pole-zero cancellations in the product $L(s)P_{22}(s)R(s)$ or, if not, its result would be improper and we then need a polynomial $T(s)$ to remove its polynomial part which means that Corollary 2.1 does not apply. Internal properness, however, is not required for all applications (PID controllers do not, in general, properize), in which case we can just use Corollary 2.2. The following is an example from [7] where the optimal solution is constructed without internal properness requirement.

Example 3 Let us consider the singular H_∞ control problem:

$$P(s) = \begin{pmatrix} 1 & \frac{-1+s}{6-5s+s^2} \\ 1 & \frac{-1+s}{6-5s+s^2} \end{pmatrix}.$$

After multiplication of its second column by $R(s) = s+1$ we get:

$$\hat{P}(s) = \begin{pmatrix} 1 & \frac{-1+s^2}{6-5s+s^2} \\ 1 & \frac{-1+s^2}{6-5s+s^2} \end{pmatrix}.$$

The optimal central controller $\hat{K}(s)$ for $\hat{P}(s)$ is found for $\gamma = 6$:

$$\hat{K}(s) = \frac{-30+5s}{6(1+s)}$$

which gives:

$$K(s) = \frac{-30+5s}{6}.$$

The same result is obtained in [7] using the polynomial approach. Note that in this example, internal properness is violated (in particular, $K(s)/(1 - K(s)P_{22}(s))$ is not proper).

In the following example, we show how compensators can be used not only to remove poles and zeros at infinity but also zeros on the $j\omega$-axis which violate the usual assumptions for the application of the Riccati-based solution method. In this paper we do not present a systematic way for designing compensators that remove these offending zeros, for that, we refer interested readers to [1].

Example 4 Consider the mixed-sensitivity problem introduced in [6]:

$$\left\| \begin{array}{c} w_1(I-GK)^{-1} \\ w_3GK(I-GK)^{-1} \end{array} \right\|_\infty < 1$$

where

$$G(s) = \frac{-4.444e^5 + 4.007e^2 s + 5.498s^2}{1.214e^5 + 9.520e^3 s + 93.72s^2 + s^3},$$

$$w_1(s) = \frac{20\rho}{s(s+20)}$$

and

$$w_3(s) = \frac{(s+30)(s+60)}{18000}.$$

The standard plant associated with this mixed-sensitivity formulation is the following:

$$P(s) = \left[\begin{array}{c|c} w_1(s) & -w_1(s)G(s) \\ 0 & w_3(s)G(s) \\ \hline I & -G(s) \end{array} \right].$$

It is not difficult to see that this plant is not normalized ($w_3(s)G(s)$ is not proper), and, moreover, P_{21} has a zero at zero.

To solve this problem, we start by normalizing $P(s)$ using the pre-compensator:

$$R(s) = 1/(s+1).$$

Thanks to Corollary 2.1, we can also introduce the post-compensator

$$L(s) = (s+1)/s$$

which removes the zero at zero of P_{21}. The result is the standard problem

$$\hat{P}(s) = \left[\begin{array}{c|c} w_1 & -w_1G/(s+1) \\ 0 & w_3G/(s+1) \\ \hline (s+1)/s & -G/s \end{array} \right]$$

to which we can directly apply the Riccati-based solution.

The optimal central solution to this problem (for $\rho = 1$) is

$$\hat{K}(s) = \frac{-39667944 - 3111342.5s - 30608.223s^2 - 326.88s^3}{8419382.8 + 135783.82s + 662.59s^2 + s^3}.$$

The optimal controller $K(s)$ is then obtained as follows:

$$K(s) = R(s)\hat{K}(s)L(s) = \hat{K}(s)/s.$$

As expected, $K(s)$ contains a pole at zero and is strictly proper.

3 Basic PSSD operations

Basic operations such as concatenation, addition, multiplication, inversion and linear fractional transformations on system transfer functions are much more easier to implement in state-space domain. That is because there exist explicit expressions for the state-space description of the results of these basic operations in terms of the state-space descriptions of their arguments, guarantying "generic" minimality.

In this section, we present algorithms for performing some basic operations on PSSD's: concatenation, addition and multiplication. These operations are needed for the development of the normalization procedure presented in the next section. For system inversion and feedback operations see [9].

Let $\Sigma_1 = (A_1, B_1, C_1, D_1(s))$ and $\Sigma_2 = (A_2, B_2, C_2, D_2(s))$ be respectively PSSD's realizing $\Sigma_1(s)$ and $\Sigma_2(s)$. The concatenation operations consist in constructing the PSSD's,

$$\Sigma_r \triangleq \begin{bmatrix} \Sigma_1 & \Sigma_2 \end{bmatrix}$$

realizing

$$\Sigma_r(s) = \begin{bmatrix} \Sigma_1(s) & \Sigma_2(s) \end{bmatrix}$$

and

$$\Sigma_c \triangleq \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix}$$

realizing

$$\Sigma_c(s) = \begin{bmatrix} \Sigma_1(s) \\ \Sigma_2(s) \end{bmatrix}.$$

These realizations can easily be constructed as follows:

$$\Sigma_r(s) = \left(\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, \begin{bmatrix} D_1(s) \\ D_2(s) \end{bmatrix} \right).$$

and

$$\Sigma_c(s) = \left(\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}, \begin{bmatrix} D_1(s) & D_2(s) \end{bmatrix} \right).$$

For the addition operation, assuming that Σ_1 and Σ_2 have compatible dimensions, a realization of

$$\Sigma_a(s) = \Sigma_1(s) + \Sigma_2(s)$$

is given by

$$\Sigma_a = \left(\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, \begin{bmatrix} D_1(s) + D_2(s) \end{bmatrix} \right).$$

We denote the PSSD addition operation simply with a plus, i.e., $\Sigma_a = \Sigma_1 + \Sigma_2$.

Unlike the two previous cases, the results of standard state-space descriptions cannot trivially be extended to the case of PSSD's as far as system multiplication is concerned, i.e., when we like to construct the PSSD, Σ, realizing

$$\Sigma(s) = \Sigma_1(s)\Sigma_2(s).$$

We denote the PSSD multiplication operation with a star:

$$\Sigma = \Sigma_1 \star \Sigma_2.$$

Theorem 3.1 *Let $\Sigma = \Sigma_1 \star \Sigma_2$. Then a PSSD of $\Sigma = (A, B, C, D(s))$ can be constructed as follows:*

$$A = \begin{bmatrix} A_1 & B_1C_2 \\ 0 & A_2 \end{bmatrix}, \quad B = Y, \quad C = T,$$

and

$$D(s) = \begin{bmatrix} C_1 & D_1(s)C_2 \end{bmatrix} X(s) + Z(s)Y + D_1(s)D_2(s)$$

where the polynomial matrix $X(s)$ and the constant matrix Y satisfy

$$\begin{bmatrix} B_1D_2(s) \\ B_2 \end{bmatrix} = (sI - A)X(s) + Y$$

and where the polynomial matrix $Z(s)$ and the constant matrix T satisfy

$$\begin{bmatrix} C_1 & D_1(s)C_2 \end{bmatrix} = Z(s)(sI - A) + T.$$

The proof of existence and a method of construction for the matrices Y, T, $X(s)$ and $Z(s)$ are given in Lemma 3.1.

Proof It is straightforward to show that

$$\Sigma(s) = \begin{bmatrix} C_1 & D_1(s)C_2 \end{bmatrix} \left(sI - \begin{bmatrix} A_1 & B_1C_2 \\ 0 & A_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} B_1D_2(s) \\ B_2 \end{bmatrix} + D_1(s)D_2(s).$$

which implies that

$$\Sigma(s) = \begin{bmatrix} C_1 & D_1(s)C_2 \end{bmatrix} \left(sI - \begin{bmatrix} A_1 & B_1C_2 \\ 0 & A_2 \end{bmatrix} \right)^{-1} ((sI - A)X(s) + Y) + D_1(s)D_2(s)$$

which gives

$$\Sigma(s) = \begin{bmatrix} C_1 & D_1(s)C_2 \end{bmatrix} \left(sI - \begin{bmatrix} A_1 & B_1C_2 \\ 0 & A_2 \end{bmatrix} \right)^{-1} Y + \begin{bmatrix} C_1 & D_1(s)C_2 \end{bmatrix} X(s) + D_1(s)D_2(s)$$

which in turn implies

$$\Sigma(s) = T(sI - A)^{-1}Y + Z(s)Y + \begin{bmatrix} C_1 & D_1(s)C_2 \end{bmatrix} X(s) + D_1(s)D_2(s)$$

which proves the theorem. □

Lemma 3.1 *Let $M(s)$ be a polynomial matrix of degree n and A a square constant matrix. Then, if $M(s)$ and A have equal number of rows, there exist a polynomial matrix $N(s)$ of degree $n - 1$ and a constant matrix J such that*

$$M(s) = (sI - A)N(s) + J. \tag{3.1}$$

Proof Let

$$M(s) = \sum_{i=0}^{n} M_i s^i$$

and let N_i, $i = 0, ..., n-1$ be obtained from the following recursion

$$\begin{aligned} N_{n-1} &= M_n \\ N_{k-1} &= M_k + AN_k, \quad k = n-1, n-2, ..., 1 \end{aligned}$$

then it is straightforward to see that

$$J = M_0 + AN_0,$$

and

$$N(s) = \sum_{i=0}^{n-1} N_i s^i$$

satisfy (3.1). □

We can of course trivially modify the proof to show that $M(s)$ can also be expressed as $\hat{N}(s)(sI - A) + \hat{J}$ for a polynomial matrix $\hat{N}(s)$ and a constant matrix $\hat{J}$ (in general, $N(s) \neq \hat{N}(s)$ and $J \neq \hat{J}$). Note that this is just a simple Euclidan division. Also note that the constructive proof given for Lemma 3.1 can be used as the basis for an algorithm.

4 Construction of normalizing compensators

4.1 Regularization

Definition 4.1 *Let $\Sigma = (A, B, C, D(s))$ have a full (normal) rank transfer function $\Sigma(s)$. Then Σ is called regular if $D(s)$ is constant and has full rank.*

This means that a regular PSSD has no poles or zeros at ∞.

Theorem 4.1 *Let $\Sigma = (A, B, C, D(s))$ have a full (normal) column (resp. row) rank transfer function $\Sigma(s)$. Then there exists a square PSSD, Γ, such that*

$$\Sigma_c = \Sigma \star \Gamma \quad \textit{(resp. } \Gamma \star \Sigma\textit{)}$$

is regular.

In a sense, Γ regularizes Σ by shifting its infinite poles and zeros to other locations in the complex plane, i.e., removing the poles and zeros at infinity (we shall see later that we can choose arbitrarily the location of these shifted poles). We say that Γ is a Σ-regularizer.

Proof Without loss of generality, we shall only consider the case where $\Sigma(s)$ is full column rank. The proof is constructive and can be used as an algorithm for constructing the regularizer Γ.

The regularization process is divided into two stages: first, a PSSD, Γ_1, is constructed in such a way that

$$\Sigma_{c,1} = \Sigma \star \Gamma_1$$

is proper (i.e., it removes all the poles at infinity), second, a Γ_2 is constructed removing all the zeros at infinity of $\Sigma_{c,1}$ (i.e., $\Sigma_{c,1} \star \Gamma_2$ having no zeros at infinity). The regularizing PSSD is then given by

$$\Gamma = \Gamma_1 \star \Gamma_2.$$

We start by letting

$$\Gamma_1^{(0)} = I$$

and

$$\Sigma^{(0)} = (A^{(0)}, B^{(0)}, C^{(0)}, D^{(0)}(s)) = \Sigma$$

with

$$D^{(0)}(s) = \sum_{k=0}^{d_0} D_k^{(0)} s^k.$$

If $d_0 = 0$ (i.e., $\Sigma^{(0)}$ is proper), we are done and

$$\Gamma_1 = \Gamma_1^{(0)} = I, \quad \Sigma_{c,1} = \Sigma.$$

If $d_0 \neq 0$, find a matrix $W^{(0)}$ which column compresses $D_{d_0}^{(0)}$, i.e.,

$$D_{d_0}^{(0)} W^{(0)} = \begin{pmatrix} X^{(0)} & 0 \end{pmatrix}$$

where $X^{(0)}$ is full column rank and update Γ_1 as follows

$$\Gamma_1^{(1)} = \Gamma_1^{(0)} \star W^{(0)} \star \Lambda^{(0)} \tag{4.1}$$

where

$$\Lambda^{(0)} = \left(0, \begin{pmatrix} I & 0 \end{pmatrix}, \begin{pmatrix} I \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}\right)$$

is a PSSD realization of

$$\Lambda^{(0)}(s) = \begin{pmatrix} (1/s)I & 0 \\ 0 & I \end{pmatrix}$$

where the size of the $(1,1)$-block of $\Lambda^{(0)}(s)$ equals the number of columns of $X^{(0)}$. And update $\Sigma_{c,1}$:

$$\Sigma_{c,1}^{(1)} = \Sigma_{c,1}^{(0)} \star W^{(0)} \star \Lambda^{(0)}. \tag{4.2}$$

We can now repeat this procedure until we obtain a proper $\Sigma_{c,1}$ as follows: at stage i,

$$\Sigma_{c,1}^{(i)} = (A^{(i)}, B^{(i)}, C^{(i)}, D^{(i)}(s))$$

with

$$D^{(i)}(s) = \sum_{k=i}^{d_i} D_k^{(i)} s^k.$$

If $d_i = 0$ (i.e., $\Sigma^{(i)}$ is proper), we are done and

$$\Gamma_1 = \Gamma_1^{(i)}$$

removes all the poles at infinity and

$$\Sigma_{c,1} = \Sigma_{c,1}^{(i)}.$$

If $d_i \neq 0$, find a matrix $W^{(i)}$ which column compresses $D_{d_i}^{(i)}$, i.e.,

$$D_{d_i}^{(i)} W^{(i)} = \begin{pmatrix} X^{(i)} & 0 \end{pmatrix}$$

where $X^{(i)}$ is full column rank and update Γ_1 as follows

$$\Gamma_1^{(i+1)} = \Gamma_1^{(i)} \star W^{(i)} \star \Lambda^{(i)}$$

where

$$\Lambda^{(i)} = \left(0, \begin{pmatrix} I & 0 \end{pmatrix}, \begin{pmatrix} I \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}\right)$$

is a PSSD realization of

$$\Lambda^{(i)}(s) = \begin{pmatrix} (1/s)I & 0 \\ 0 & I \end{pmatrix}$$

where the size of the $(1,1)$-block of $\Lambda^{(i)}(s)$ equals the number of columns of $X^{(i)}$. And update $\Sigma_{c,1}$:

$$\Sigma_{c,1}^{(i+1)} = \Sigma_{c,1}^{(i)} \star W^{(i)} \star \Lambda^{(i)}.$$

Clearly, this recursion ends in a finite number of iterations, because, at each iteration the degree d_i of the polynomial part of $\Sigma_{c,1}^{(i)}$ decreases by one.

Now that we have constructed $\Sigma_{c,1}$ and Γ_1, we can start the construction of Σ_c and Γ_2. We proceed as in the previous case by initializing

$$\Gamma_2^{(0)} = I$$

and

$$\Sigma_c^{(0)} = (A^{(0)}, B^{(0)}, C^{(0)}, D^{(0)}) = \Sigma_{c,1}.$$

If $D^{(0)}$ is full rank, we are done and

$$\Gamma_2 = \Gamma_2^{(0)} = I, \quad \Sigma_c = \Sigma_{c,1}.$$

If not, find a matrix $W^{(0)}$ which column compresses $D^{(0)}$, i.e.,

$$D^{(0)} W^{(0)} = \begin{pmatrix} X^{(0)} & 0 \end{pmatrix}$$

where $X^{(0)}$ is full column rank and update Γ_2 as follows

$$\Gamma_2^{(1)} = \Gamma_2^{(0)} \star W^{(0)} \star \Lambda^{(0)}$$

where

$$\Lambda^{(0)}(s) = \begin{pmatrix} I & 0 \\ 0 & sI \end{pmatrix}$$

where the size of the $(1,1)$-block of $\Lambda^{(0)}(s)$ equals the number of columns of $X^{(0)}$. And update Σ_c:

$$\Sigma_c^{(1)} = \Sigma_c^{(0)} \star W^{(0)} \star \Lambda^{(0)}.$$

We can now repeat this procedure until we obtain a Σ_c with no zeros at infinity as follows: at stage i,

$$\Sigma_c^{(i)} = (A^{(i)}, B^{(i)}, C^{(i)}, D^{(i)}).$$

If $D^{(i)}$ is full rank, we are done and

$$\Gamma_2 = \Gamma_2^{(i)}$$

removes all zeros at infinity and

$$\Sigma_c = \Sigma_c^{(i)}.$$

If not, find a matrix $W^{(i)}$ which column compresses $D^{(i)}$, i.e.,

$$D^{(i)}W^{(i)} = \begin{pmatrix} X^{(i)} & 0 \end{pmatrix}$$

where $X^{(i)}$ is full column rank and update Γ_2 as follows

$$\Gamma_2^{(i+1)} = \Gamma_2^{(i)} \star W^{(i)} \star \Lambda^{(i)} \tag{4.3}$$

where

$$\Lambda^{(i)}(s) = \begin{pmatrix} I & 0 \\ 0 & sI \end{pmatrix}$$

where the size of the $(1,1)$-block of $\Lambda^{(i)}(s)$ equals the number of columns of $X^{(i)}$. And update Σ_c:

$$\Sigma_c^{(i+1)} = \Sigma_c^{(i)} \star W^{(i)} \star \Lambda^{(i)}. \tag{4.4}$$

This second stage of the algorithm is nothing but Silverman's structure algorithm [10].[2] The algorithm converges in less than n steps where n denotes the order of $\Sigma_{c,1}$. This completes the proof. □

Remarks:

1- All the operations involved in the regularization algorithm are performed in the state-space domain, in particular (4.1), (4.2), (4.3), and (4.4). Note that the result of (4.2) should be a reduction in the degree of the polynomial part of $\Sigma_{c,1}$. Due to numerical round-off errors, however, we may end up with a result having the same polynomial degree as the previous $\Sigma_{c,1}$, the coefficient of the highest degree of its polynomial part being very small. This coefficient should be set to zero otherwise the first stage of the algorithm may never end.

2- For the sake of numerical robustness, in the actual implementation of the algorithm, all the column compressions are performed using orthogonal matrices.

3- We can alter the second stage of the regularization algorithm in such a way as to construct Γ directly (instead of Γ_2) by taking as initial condition $\Gamma_2^{(0)} = \Gamma_1$.

4- All the finite poles and the zeros of $\Gamma(s)$, as constructed in the above algorithm, are at zero. They can however easily be arbitrarily assigned to α. In order to construct a regularizer for Σ having its finite poles and zeros at α, find a regularizer Γ_α for

$$\Sigma_\alpha = ((A - \alpha I), B, C, D(s + \alpha))$$

having finite poles and zeros at zero (using the above algorithm). Then if $\Gamma_\alpha = (A_g, B_g, C_g, D_g(s))$,

$$\Gamma = (A_g + \alpha I, B_g, C_g, D_g(s - \alpha))$$

is a regularizer for Σ having its finite poles and zeros at α.

[2]Note that the PSSD multiplication has allowed us to write this algorithm in a simpler fashion.

4.2 Compensator construction

The regularization algorithm presented in the previous section can be used for the construction of pre- and post-compensators R and L. This can be done simply by letting R be a regularizer for P_{12} and L a regularizer for P_{21} having finite poles and zeros at some $\alpha < 0$. Then the $(1,2)$ and $(2,1)$ blocks of

$$P_{tmp} = \begin{pmatrix} I & 0 \\ 0 & L \end{pmatrix} \star P \star \begin{pmatrix} I & 0 \\ 0 & R \end{pmatrix} \tag{4.5}$$

have no zeros or poles at infinity.[3]

Even though, (4.5) can be used to construct P_{tmp}, for numerical robustness it may be preferable to construct it step by step during the regularization processes. This can be done by updating P, and not just P_{12} and P_{21}, at every step of the regularization algorithm.

Let

$$P_{tmp} = \left[\begin{array}{c|cc} A_{tmp} & B_{1tmp} & B_{2tmp} \\ \hline C_{1tmp} & D_{11tmp} & D_{12tmp} \\ C_{2tmp} & D_{21tmp} & D_{22tmp}(s) \end{array}\right],$$

then clearly, we can take

$$T(s) = D_{22tmp}(s)$$

which yields

$$\hat{P} = \left[\begin{array}{c|cc} A_{tmp} & B_{1tmp} & B_{2tmp} \\ \hline C_{1tmp} & D_{11tmp} & D_{12tmp} \\ C_{2tmp} & D_{21tmp} & 0 \end{array}\right]$$

where D_{12tmp} and D_{21tmp} have respectively full column and full row rank; this means that $\hat{P}$ is normalized. Assuming that other conditions are satisfied, Riccati-based solution method can be used to construct $\hat{K}(s)$ and consequently $K(s)$ using (2.5). The expression in (2.5) can be evaluated very effectively using PSSD basic operations [9].

Example 5 Consider the following H_∞ standard problem

$$P(s) = \left(\begin{array}{c|cc} \frac{1}{2+s} & -1+s & 0 \\ 0 & 1 & \frac{1}{1+s} \\ \hline 1+s & \frac{-1+s}{3+s} & 0 \end{array}\right)$$

which admits the following PSSD:

$$P = \left(\begin{array}{c|c} A & B \\ \hline C & D(s) \end{array}\right) = \left(\begin{array}{c|c} \begin{pmatrix} -3 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{pmatrix} & \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & -2 \\ -2 & 0 & 0 \end{pmatrix} \\ \hline \begin{pmatrix} 0 & 0 & -1/2 \\ 0 & -1/2 & 0 \\ -2 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & -1+s & 0 \\ 0 & 1 & 0 \\ 1+s & 1 & 0 \end{pmatrix} \end{array}\right).$$

The PSSD's of the $(1,2)$ and $(2,1)$ blocks of P are respectively:

$$P_{12} = \left(\begin{array}{c|c} \begin{pmatrix} -3 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{pmatrix} & \begin{pmatrix} 2 & 0 \\ 0 & -2 \\ 0 & 0 \end{pmatrix} \\ \hline \begin{pmatrix} 0 & 0 & -1/2 \\ 0 & -1/2 & 0 \end{pmatrix} & \begin{pmatrix} -1+s & 0 \\ 1 & 0 \end{pmatrix} \end{array}\right)$$

[3]Due to numerical round-off errors. $(1,2)$ and $(2,1)$ blocks of P_{tmp} may contain residual polynomial parts. They can be set to zero because we know, a priori, that they are due to numerical errors.

and

$$P_{21} = \left(\begin{array}{c|c} \begin{pmatrix} -3 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix} \\ \hline \begin{pmatrix} -2 & 0 & 0 \end{pmatrix} & 1+s \end{array} \right).$$

By applying the normalization algorithm (with $\alpha = 1$) we obtain:

$$R = \left(\begin{array}{c|c} -1 & \begin{pmatrix} 0 & 1 \end{pmatrix} \\ \hline \begin{pmatrix} -1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 1+s & 0 \end{pmatrix} \end{array} \right)$$

which has the following transfer function:

$$R(s) = \begin{pmatrix} 0 & \frac{-1}{1+s} \\ 1+s & 0 \end{pmatrix}$$

and

$$L = \left(\begin{array}{c|c} -1 & 1 \\ \hline 1 & 0 \end{array} \right)$$

which has the following transfer function

$$L(s) = \frac{1}{1+s}.$$

The corresponding normalized plant is

$$\hat{P} = \left(\begin{array}{c|c} \begin{pmatrix} -1 & -2 & 0 & 0 & -1 \\ 0 & -3 & 0 & 0 & -2 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \hline \begin{pmatrix} 0 & 0 & 0 & -1/2 & 2 \\ 0 & 0 & -1/2 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array} \right)$$

which has the following transfer function:

$$\hat{P}(s) = \begin{pmatrix} \frac{1}{2+s} & 0 & \frac{1-s}{1+s} \\ 0 & 1 & \frac{-1}{1+s} \\ 1 & 0 & \frac{1-s}{(s+1)^2(s+3)} \end{pmatrix}.$$

The optimal central controller for this normalized plant (using Riccati-based solution method) is

$$\hat{K}(s) = \frac{1}{17 + 34s + 18s^2 + 3s^3} \begin{pmatrix} -(s+1)(s+3) \\ -(s+1)^2(s+3) \end{pmatrix}$$

with $\gamma_{opt} = 1/3$. The solution to the original problem is then given by:

$$K(s) = R(s)\hat{K}(s)L(s) = \frac{1}{17 + 34s + 18s^2 + 3s^3} \begin{pmatrix} 3+s \\ -(s+1)(s+3) \end{pmatrix}.$$

5 Conclusion

The method presented in this paper enlarges the class of H_∞ standard problems to which Riccati-based solution method [2] can be applied, problems which until now could have only been solved using the polynomial approach [7]. This method applies to H_2 standard problems equally well since Riccati-based solution to this problem requires the same type of pole-zero structure at infinity. The normalization procedure, developed in this paper, can also be used to solve the inner-outer factorization problem of improper rational matrices. We shall discuss this in a future paper.

References

[1] Copeland, B. R. and Safonov, M. G., (1992), Zero cancelling compensators for singular control problems and their application to the inner-outer factorization problem, Int. J. Robust and Nonlinear Control, 2.

[2] Doyle J. C. , Glover K., Khargonekar P., Francis B.A., (1989), State-Space Solutions to Standard H_2 and H_∞ Control Problems, IEEE Trans. Auto. Contr., 34.

[3] Francis B., (1987), A course in H_∞ control theory, Springer Verlag, Berlin.

[4] Glover K., Doyle J.C., (1988), State-Space Formulae for all Stabilizing Controllers that Satisfy an H_∞-norm Bound and Relations to Risk Sensitivity, Syst. Contr. Letters, 11.

[5] Kailath T., (1980), Linear Systems, Prentice Hall, Englewood Cliffs, NJ.

[6] Kuraoka H., Ohka N., Hosoe S., Zhang F., (1990), Application of H_∞ Design to Automative Fuel Design, IEEE Control System Magazine.

[7] Kwakernaak, H., (1990), The polynomial approach to H_∞-optimal regulation. Lecture notes in Mathematics, 1990 CIME course on recent developments in H_∞ control theory.

[8] Nikoukhah, R. and Delebecque, F., On the design of H_∞ controllers when standard assumptions are not satisfied. *submitted to ACC-93.*

[9] Nikoukhah, R. and Delebecque, F., (1992), State-space representation of rational matrices in Ψlab, Inria report.

[10] Silverman, L. M., (1969), Inversion of multivariable linear systems. *IEEE Trans. Aut. Control*, AC-14.

[11] Stoorvogel, A. A., (1991), The singular H_∞ control problem with dynamic measurement feedback, SIAM J. Opt. and Control, 29.

ISSN 0249-6399