Discrete Robust SPR Design via Semidefinite Programming

Didier Henrion[1,2]

Abstract

We show that the problem of designing a discrete-time fixed-order controller that renders a given set of scalar transfer functions simultaneously strictly positive real (SPR) can be cast into a convex semidefinite programming problem. The extension to strengthened robust SPR design is also covered. A Matlab/SeDuMi implementation of the design method is provided and successfully applied to numerical examples found in the literature.

Keywords

Linear systems, Discrete-time, Strictly positive real, Polynomial, Semidefinite programming.

1 Introduction

Strictly positive real (SPR) transfer functions have several applications in control and signal processing. They are used to ensure convergence of recursive identification algorithms and filtered-error adaptive algorithms, see e.g. [Dasgupta, 1987, Anderson, 1990, Anderson, 1994, Mosquera, 1999] and the references therein. SPR functions also arise when studying stability of non-linear systems within the Lur'e-Popov framework [Šiljak, 1998]. When system parameters are uncertain, a problem of central importance is that of designing a controller so that a whole family of transfer functions be robustly SPR. This is the problem we will address in this paper.

We consider a discrete-time scalar linear system with monic characteristic polynomial

$$a(z) = a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}, \quad a_0 = 1$$

[1]Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique, 7 Avenue du Colonel Roche, 31 077 Toulouse, cedex 4, France. Fax: +33 5 61 33 69 69. E-mail: `henrion@laas.fr`

[2]Also with the Institute of Information Theory and Automation, Czech Academy of Sciences, Pod vodárenskou věží 4, 182 08 Praha, Czech Republic.

of degree n in the backward shift operator z^{-1} [Kučera, 1979]. We consider that polynomial $a(z)$ is uncertain and belongs to a family which is a convex polytope with N given vertices $a_1(z), \ldots, a_N(z)$, see [Dasgupta, 1987]. It is assumed that the whole polytope is Schur stable, i.e. all its polynomial elements have roots with magnitude strictly greater than 1. The problem of robust SPR design can then be formulated as follows: find a monic polynomial

$$c(z) = c_0 + c_1 z^{-1} + \cdots + c_m z^{-m}, \quad c_0 = 1$$

of degree m such that the N scalar transfer functions $c(z)/a_k(z)$ are simultaneously SPR for all $k = 1, \ldots, N$. SPRness means that the transfer functions must be stable, causal ($m \leq n$) and with real parts positive along the unit circle, i.e.

$$\mathrm{Re} \frac{c(e^{jw})}{a_k(e^{jw})} > 0$$

for all real w and $k = 1, \ldots, N$.

A necessary and sufficient condition of existence of such a controller polynomial $c(z)$ is given in [Anderson, 1990] as a phase inequality condition. The condition is not constructive, except in the case of transfer functions of low order. Moreover, it does not provide any bound on the degree of the solution. The problem has been revisited recently in [Mosquera, 2001] in the special case of $N = 2$ polynomials. A constructive algebraic design procedure is given, based on interpolation of rational functions. The procedure allows to bound a priori the degree of $c(z)$.

In this paper we propose a fully general, yet straightforward robust SPR design procedure based on semidefinite programming. It is valid for any number N of vertex polynomials. A necessary and sufficient condition for the existence of a polynomial $c(z)$ of given degree m that solves the discrete-time robust SPR design problem is formulated as a linear inequality over the cone of positive semidefinite matrices. This is a convex optimization problem ubiquitous in control and signal processing [Vandenberghe, 1996] for which several efficient solvers are now available. The approach is flexible enough to address the strengthened robust SPR problem [Anderson, 1994], defined as the problem of finding $c(z)$ such that, given a scalar γ, the transfer functions $c(z)/a_k(z) - \gamma$ are simultaneously SPR for all $k = 1, \ldots, N$. In its original formulation, our semidefinite programming approach yields a finite-impulse response (FIR) controller polynomial $c(z)$ of given degree, but it can be modified so as to provide an infinite-impulse response (IIR) controller transfer function with given numerator or denominator. Last but not least, additional convex constraints on the controller coefficients can freely be incorporated, such as spectral or interpolation constraints on the controller dynamics.

The simplicity and efficiency of the approach is illustrated on several numerical examples found in the literature, and a basic Matlab[1]/SeDuMi [Sturm, 1999] implementation of the robust SPR design algorithm is given in the Appendix. A more sophisticated version of the design algorithm will be implemented into the next release 3.0 of the Polynomial Toolbox for Matlab [Polyx Ltd., 2001].

[1]Matlab is a trademark of The MathWorks, Inc.

2 Main Result

We go straight to the point and state our main result which provides a FIR solution of given degree to the robust SPR design problem.

Theorem 1 *Suppose we are given a stable polytope of discrete-time polynomials whose monic vertices of degree n are*

$$a_k(z) = a_{k0} + a_{k1}z^{-1} + \cdots + a_{kn}z^{-n}, \quad a_{k0} = 1$$

for $k = 1, \ldots, N$. Then there exists a monic FIR controller polynomial

$$c(z) = c_0 + c_1 z^{-1} + \cdots + c_m z^{-m}, \quad c_0 = 1$$

of given degree m that makes the N scalar transfer functions $c(z)/a_k(z)$ simultaneously SPR if and only if there exist N symmetric positive semidefinite matrices P_k with entries $p_{k_{ij}}$, $i, j = 0, 1, \ldots, n$ such that

$$\sum_{i-j=l} a_{k_i} c_j = \sum_{i-j=l} p_{k_{ij}}, \quad l = 1, \ldots, n$$

and

$$\sum_i a_{k_i} c_i = \sum_i p_{k_{ii}} - \epsilon_k$$

for given scalars $\epsilon_k > 0$ and $k = 1, \ldots, N$.

Proof The proof is split into two parts. First, we show that the discrete-time SPR design problem amounts to checking positivity of some trigonometric polynomial. Second, we show that the positivity condition can be expressed as a semidefinite programming problem as above.

In [Anderson, 1994] it was shown that SPRness of each vertex transfer function $c(z)/a_k(z)$ for $k = 1, \ldots, N$ is necessary and sufficient to ensure SPRness of the whole polytope. So in the development below, we will focus on the problem of ensuring positive realness of only one transfer function $c(z)/a(z)$.

Strict positive realness of the transfer function $c(z)/a(z)$ is equivalent to the requirement that

$$\mathrm{Re}\frac{c(e^{jw})}{a(e^{jw})} = \frac{1}{2}\left(\frac{c(e^{jw})}{a(e^{jw})} + \frac{c(e^{-jw})}{a(e^{-jw})}\right) = \frac{1}{2}\frac{a(e^{-jw})c(e^{jw}) + a(e^{jw})c(e^{-jw})}{a(e^{-jw})a(e^{jw})} > 0$$

for all real w. Since each polynomial $a(z)$ is assumed to be stable, the denominator never vanishes and the SPRness requirement is equivalent to the positivity requirement

$$\begin{aligned} p(w) &= a(e^{-jw})c(e^{jw}) + a(e^{jw})c(e^{-jw}) \\ &= p_n e^{-njw} + \cdots + p_1 e^{-jw} + 2p_0 + p_1 e^{jw} + \cdots + p_n e^{njw} \\ &= 2\textstyle\sum_{l=0}^{n} p_l \cos(lw) \qquad > 0 \end{aligned}$$

on the trigonometric polynomial $p(w)$ for all real w. The idea of reformulating a positive realness condition as a polynomial positivity condition originates from [Šiljak, 1998]. Identification of like powers of e^{jw} readily gives the relationship

$$p_l = \sum_{i-j=l} a_i c_j, \quad l = 0, \ldots, n \tag{1}$$

between coefficients of $p(w)$, plant coefficients of $a(z)$ and controller coefficients of $c(z)$. Linear equation (1) can be formulated in a more transparent way as

$$\begin{bmatrix} 2p_0 \\ p_1 \\ \vdots \\ p_n \end{bmatrix} = \left\{ \begin{bmatrix} a_0 & a_1 & \cdots & a_n \\ a_1 & \cdots & a_n & 0 \\ \vdots & & & \vdots \\ a_n & 0 & \cdots & 0 \end{bmatrix} + \begin{bmatrix} a_0 & a_1 & \cdots & a_n \\ 0 & a_0 & \cdots & a_{n-1} \\ \vdots & & & \vdots \\ 0 & \cdots & 0 & a_0 \end{bmatrix} \right\} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

where leading coefficients $c_{m+1}, \ldots, c_n$ are set to zero. The matrix appearing in the above equation is usually referred to as the Jury matrix of polynomial $a(z)$ [Jury, 1974]. It is non-singular when $a(z)$ is Schur stable. Thus so far we have shown that SPRness of a transfer function with sought numerator polynomial $c(z)$ is equivalent to positiveness of a trigonometric polynomial $p(w)$ whose coefficients depend linearly on the coefficients of $c(z)$.

Now we proceed by showing that positivity of $p(w)$ can be formulated as the existence of a positive semidefinite matrix satisfying some linear equalities. First note that $p(w) > 0$ if and only if $p(w) \geq \epsilon$ for some given scalar $\epsilon > 0$, so that we can equivalently focus on the non-strict positivity condition.

The trigonometric positivity condition

$$p(w) = 2 \sum_{l=0}^{n} p_l \cos(lw) \geq 0 \tag{2}$$

is satisfied for all real w if and only if there exists a symmetric positive semidefinite matrix P with entries p_{ij}, $i, j = 0, 1, \ldots, n$ such that

$$p_l = \sum_{i-j=l} p_{ij}, \quad l = 0, 1, \ldots, n. \tag{3}$$

To see this, note that P positive semidefinite obviously implies

$$\frac{1}{2} p(w) = \begin{bmatrix} 1 \\ e^{-jw} \\ \vdots \\ e^{-njw} \end{bmatrix}^T \begin{bmatrix} p_{00} & p_{10} & \cdots & p_{n0} \\ p_{10} & p_{11} & \cdots & p_{n1} \\ \vdots & & & \vdots \\ p_{n0} & p_{n1} & \cdots & p_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ e^{jw} \\ \vdots \\ e^{njw} \end{bmatrix} \geq 0$$

hence positivity of $p(w)$. Conversely, $p(w) \geq 0$ implies that there exists a decomposition of polynomial $p(w)$ as a sum of squares, see [Nesterov, 2000, Genin, 2000b, Alkire, 2001] for a proof based on the conic duality, see [Lasserre, 2001] for a proof based on the theory of moments, see [Parrilo, 2000] for references on related semialgebraic problems such as

Hilbert's 17th problem. In other words, we can write $p(w) = \sum_k q_k^2(w)$ where polynomials $q_k(w) = q_{k0} + q_{k1}w + \cdots$ have degree at most n. Then, we can check that the positive semidefinite matrix P with entries $p_{ij} = \sum_k q_{k_i} q_{k_j}$ satisfies the required linear equalities (3).

Finally, if we combine linear equalities (1) and the N positivity conditions (2) on each vertex polynomial $p_k(w)$ corresponding to each denominator vertex $a_k(z)$, we obtain the required semidefinite programming conditions of Theorem 1. $\square$

The main strength of Theorem 1 is in showing that the fixed-order robust SPR design problem can be formulated as the problem of finding a set of scalars and positive semidefinite matrices satisfying some linear equalities. This is a convex optimization problem arising frequently in control and signal processing [Vandenberghe, 1996] but also in convex relaxations to combinatorial optimization problems [Lasserre, 2001]. Several user friendly packages have been developed to solve this convex optimization problem, such as the freeware SeDuMi interfaced with Matlab [Sturm, 1999]. In the next section, we show how robust SPR design problems can be solved in a matter of seconds with the help of this software.

Before doing this, we will show that our approach is also suitable for solving the related problem of strengthened SPR design [Anderson, 1994].

Corollary 1 *Under the same assumptions as in Theorem 1, and given a real scalar γ, there exists a monic FIR controller polynomial*

$$c(z) = c_0 + c_1 z^{-1} + \cdots + c_m z^{-m}, \quad c_0 = 1$$

of given degree m that makes the N scalar transfer functions $c(z)/a_k(z) - \gamma$ simultaneously SPR if and only if there exist N symmetric positive semidefinite matrices P_k with entries $p_{k_{ij}}$, $i, j = 0, 1, \ldots, n$ such that

$$\sum_{i-j=l} a_{k_i} c_j = \sum_{i-j=l} p_{k_{ij}} + 2\gamma \sum_{i-j=l} a_{k_i} a_{k_j}, \quad l = 1, \ldots, n$$

and

$$\sum_i a_{k_i} c_i = \sum_i p_{k_{ii}} + 2\gamma \sum_i a_{k_i}^2 - \epsilon_k$$

for given scalars $\epsilon_k > 0$ and $k = 1, \ldots, N$.

Proof The proof follows easily from the proof of Theorem 1 as soon as it is noticed that

$$\mathrm{Re} \frac{c(e^{jw})}{a(e^{jw})} = \frac{1}{2} \frac{a(e^{-jw})c(e^{jw}) + a(e^{jw})c(e^{-jw})}{a(e^{-jw})a(e^{jw})} > \gamma$$

if and only if

$$p(w) = a(e^{-jw})c(e^{jw}) + a(e^{jw})c(e^{-jw}) - 2\gamma a(e^{-jw})a(e^{jw})$$

is a positive trigonometric polynomial. $\square$

Similarly, if $\gamma = 0$ and the sought controller $c(z)$ is not expressed as an FIR filter but rather as an IIR transfer function

$$c(z) = \frac{c_n(z)}{c_d(z)} = \frac{c_{n0} + c_{n1}z^{-1} + \cdots + c_{nm}z^{-m}}{c_{d0} + c_{d1}z^{-1} + \cdots + c_{dm}z^{-m}}$$

we can pursue the same approach to come up with a positivity requirement on the trigonometric polynomial

$$p(w) = a(e^{-jw})c_d(e^{-jw})c_n(e^{jw}) + a(e^{jw})c_d(e^{jw})c_n(e^{-jw}).$$

The semidefinite programming condition becomes then bilinear thus non-convex in coefficients of sought numerator and denominator polynomials $c_n(z)$ and $c_d(z)$. Linearity and convexity are retrieved as soon as we assume that either $c_n(z)$ or $c_d(z)$ is given. For example, setting $c_d(z)$ from the outset amounts to specifying the dynamics of the IIR controller.

Finally, let us point out the fact that the semidefinite programming formulation of Theorem 1 allows for incorporating additional convex constraints on controller coefficients c_k. For example, one can think about imposing that the FIR controller $c(z)$ has a zero at $z^{-1} = x$ where x is a given number, with the help of the linear interpolation condition $1 + \sum_{k=1}^{m} x^k c_k = 0$. We can also enforce some coefficient in $c(z)$ to zero, this may simplify the implementation of the controller.

3 Numerical Examples

The software SeDuMi [Sturm, 1999] allows to solve convex optimization problems over self-dual cones, including the cone of positive semidefinite matrices. We implemented a Matlab macro that performs fixed-order strengthened robust SPR design as in Corollary 1 with the help of the freeware SeDuMi. The full listing of the macro is given in the Appendix. The macro was used to solve the numerical examples described below.

3.1 First Example

It is taken from Example 10 in [Mosquera, 2001]. A FIR controller polynomial $c(z)$ of degree $m = 3$ is sought that makes simultaneously SPR the $N = 2$ polynomials

$$a_1(z) = (1 - 0.3z^{-1})^3, \quad a_2(z) = (1 - 0.9z^{-1})^3$$

of degree $n = 3$. With input

```
Ac = [1.0000 -0.9000 0.2700 -0.0270; 1.0000 -2.7000 2.4300 -0.7290]
```

our macro called with the Matlab instruction `c = robuspr(Ac)` returns after 0.7 seconds of CPU time on a Sun Ultra 5 workstation

```
c = [1.0000 -0.7950 -0.8559 0.6909]
```

which corresponds to the controller polynomial

$$c(z) = 1 - 0.7950z^{-1} - 0.8559z^{-2} + 0.6909z^{-3}.$$

On Figure 1 we represent the graphs of the real parts of the two vertex transfer functions $c(z)/a_1(z)$ and $c(z)/a_2(z)$ as functions of phase $0 \leq w \leq \pi$. We can check that both transfer functions are indeed SPR. Consequently, the whole segment $\lambda c(z)/a_1(z) + (1-\lambda)c(z)/a_2(z)$ is robustly SPR for all $0 \leq \lambda \leq 1$.

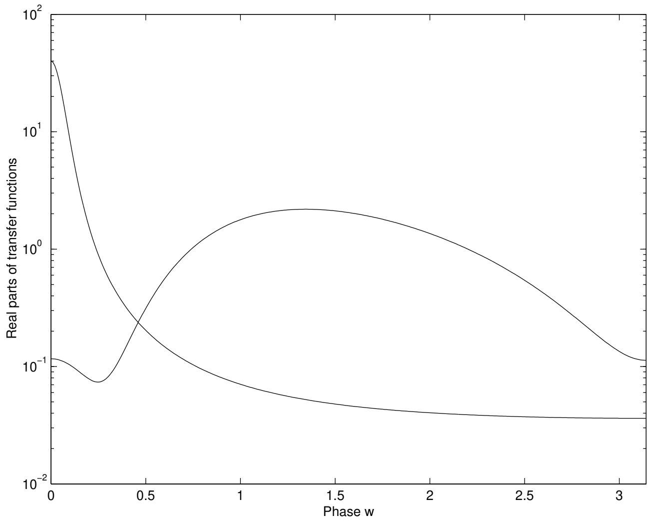

Figure 1: Real parts of the 2 transfer functions of 3rd order for a 3rd order controller.

If we are seeking a controller of order $m = 3$ that ensures strengthened robust SPRness with a margin of e.g. $\gamma = 0.1$, we use the instruction `c = robuspr(Ac,3,0.1)` and we obtain

$$c(z) = 1 - 0.9073z^{-1} - 0.6279z^{-2} + 0.5836z^{-3}.$$

Now if we set back γ to zero and we want a polynomial $c(z)$ of degree $m = 2$, the instruction `c = robuspr(Ac,2)` returns the controller

$$c(z) = 1 - 1.6447z^{-1} + 0.6637z^{-2}$$

which renders both transfer functions SPR. Note that no second-order controller was provided in [Mosquera, 2001]. The semidefinite programming problem has no solution when $m = 1$ or $m = 0$, which indicates that the lowest order robust SPR controller in this case has FIR of degree $m = 2$.

In the semidefinite programming formulation of Theorem 1 we can freely add convex constraints on the controller coefficients. For example, if we enforce $c_2 = 0$ we obtain a robust SPR controller

$$c(z) = 1 - 1.2810z^{-1} + 0.3085z^{-3}$$

of degree $m = 3$ with zero second order coefficient. Similarly, if we enforce the two linear interpolation constraints $1 + 2c_1 + 4c_2 + 8c_3 = 0$ and $1 + 3c_1 + 9c_2 + 27c_3 = 0$ then the resulting robust SPR controller

$$c(z) = 1 - 1.8064z^{-1} + 0.9776z^{-2} - 0.1622z^{-3}$$

features two zeros at $z^{-1} = 2$ and $z^{-1} = 3$.

3.2 Second Example

We consider Example 1 in [Mosquera, 1999], where a controller of degree $m = 3$ is sought that renders simultaneously SPR $N = 3$ polynomials of degree $n = 16$:

$$\begin{aligned} a_1(z) &= (1 - 0.1z^{-1})^4(1 + 0.3z^{-1})^4(1 + 0.7z^{-1})^4(1 - 0.4z^{-1} + 0.29z^{-2})^2 \\ a_2(z) &= (1 - 0.2z^{-1})^4(1 + 0.4z^{-1})^4(1 + 0.5z^{-1})^4(1 - 0.2z^{-1} + 0.5z^{-2})^2 \\ a_3(z) &= (1 - 0.3z^{-1})^4(1 + 0.1z^{-1})^4(1 + 0.6z^{-1})^4(1 + 0.8z^{-1} + 0.32z^{-2})^2. \end{aligned}$$

With the macro `robuspr` we obtain after less than 2 seconds of CPU time on a Sun Ultra 5 workstation the controller polynomial

$$c(z) = 1 + 1.9364z^{-1} + 1.3339z^{-2} + 0.3312z^{-3}.$$

In [Mosquera, 1999] a controller of the same order was found with a more complicated approach based on appropriate frequency gridding, weighted least-squares and linear programming. On Figure 2 we notice that the three vertex transfer functions are indeed SPR, which implies robust SPRness of the whole polytope. We can check with macro

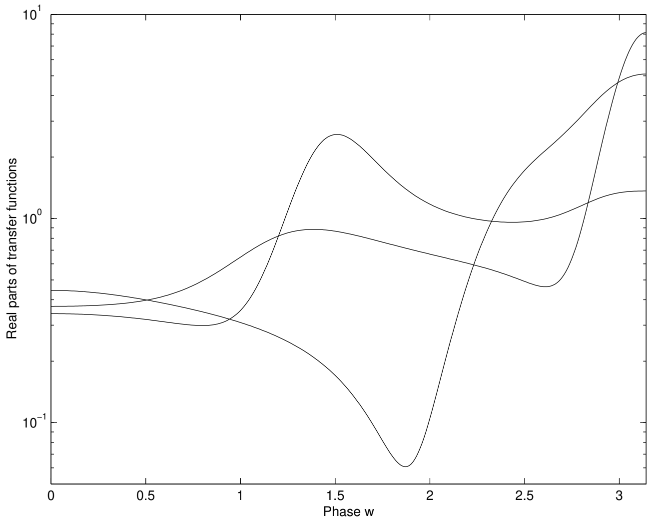

Figure 2: Real parts of the 3 transfer functions of 16th order for a 3rd order controller.

`robuspr` that there is no FIR controller of degree $m < 3$ that renders the whole polytope of polynomials SPR, which is consistent with the results obtained in [Mosquera, 1999].

Finally, as in Example 2 in [Mosquera, 1999] if we add four roots to the three polynomials of the previous example:

$$\begin{aligned}\hat{a}_1(z) &= a_1(z)(1 - z^{-1} + 0.41z^{-2})^2 \\ \hat{a}_2(z) &= a_2(z)(1 - 0.4z^{-1} + 0.4z^{-2})^2 \\ \hat{a}_3(z) &= a_3(z)(1 - 0.8z^{-1} + 0.65z^{-2})^2\end{aligned}$$

then we obtain a lowest order FIR controller

$$c(z) = 1 + 0.8422z^{-1} + 0.4632z^{-2} + 0.4987z^{-3}$$

of degree $m = 3$. Graphs of the three SPR transfer functions of order $n = 20$ are given on Figure 3.

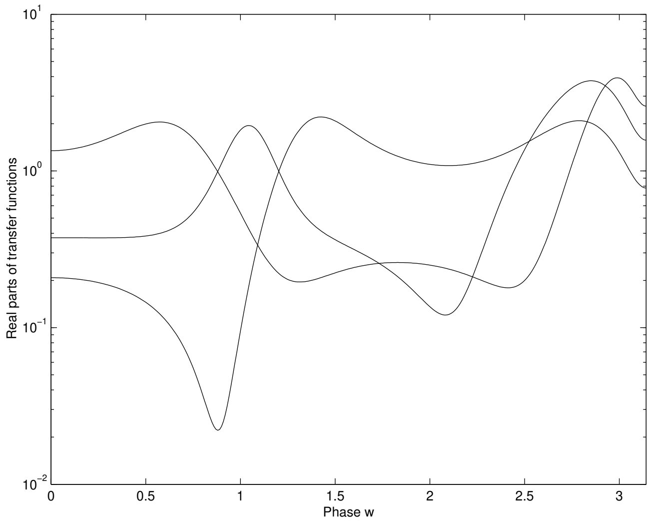

Figure 3: Real parts of the 3 transfer functions of 20th order for a 3rd order controller.

4 Conclusion

We have proposed a straightforward, fully general approach to design a discrete-time FIR controller that makes a finite set of given transfer functions strictly SPR. It is based on semidefinite programming, for which several efficient solvers are now available.

The approach is not limited to discrete-time scalar systems. It was applied to continuous-time scalar systems in [Henrion, 2000]. Extension to multivariable systems is also possible, since semidefinite programming conditions are also valid for assessing positivity of a polynomial matrix [Genin, 2000a].

Further work must be devoted to develop numerical algorithms for solving polynomial positivity conditions formulated as semidefinite programming problems. Some directions for improving numerical stability with alternative polynomial bases (such a Chebyshev polynomials) are mentioned in [Genin, 2000b]. A new algorithm with better computational complexity is proposed in [Alkire, 2001] that takes advantage of the special Toeplitz structure of the dual formulation.

Acknowledgment

I would like to thank Jean-Bernard Lasserre, LAAS-CNRS, Toulouse, for providing preprints of the papers [Nesterov, 2000] and [Lasserre, 2001]. I am also grateful to Dragoslav D. Šiljak, University of Santa Clara, California, for providing a copy of the paper [Šiljak, 1998].

Appendix

We provide a Matlab/SeDuMi implementation of the semidefinite programming condition of Corollary 1. If we set $\gamma = 0$ then we recover the condition of Theorem 1. The M-file is available for downloading at the address

`http://www.laas.fr/~henrion/Software/robuspr.m`

The macro works with the versions 5.3 and 6.0 of Matlab and version 1.0.3 of SeDuMi. It is assumed that SeDuMi is properly installed and accessible from the Matlab prompt, following the procedure described in [Sturm, 1999].

We use SeDuMi primal standard feasibility form $Ax = b$ where vector x belongs to a self-dual cone $\mathcal{K}$. In our case $\mathcal{K}$ is the direct product of

- a quadratic Lorentz cone of dimension $m+1$ corresponding to a feasibility radius R and the controller coefficients $c_1, \ldots, c_m$ such that $R^2 \geq c_1^2 + \cdots + c_m^2$
- N cones of symmetric positive semidefinite matrices $P_1, \ldots, P_N$ of dimension $n+1$.

```
function c = robuspr(Ac,m,gamma)
% Strictly positive real design for a discrete-time polytopic system
%
% Given the coefficients
%
%   Ac = [1 a11 .. a1n; 1 a21 .. a2n; ...; 1 aN1 .. aNn]
%
% of N monic polynomials ak(z) = 1+ak1*z+..+akn*z^n, k=1,..,N of
% degree n, the command
%
%   c = ROBUSPR(Ac,m,GAMMA)
%
```

```
% computes the coefficients c = [1 c1 .. cm] of a monic polynomial
% c(z) = 1+c1*z+..+cm*z^m of given degree m such that all the transfer
% functions c(z)/ak(z) are simultaneously strictly positive real on
% the unit circle with margin GAMMA, i.e. such that the inequality
% real(c(exp(i*w))/ck(exp(i*w))) > GAMMA holds for all real w and
% k=1,..,N.
%
% The simplified syntax
%
%   c = ROBUSPR(Ac)
%
% assumes m = n and GAMMA = 0.

% Author: D. Henrion, March 15, 2001 (henrion@laas.fr).

% N = number of vertices in polytope
% n = system order
[N,n] = size(Ac); n=n-1;

% m = controller order
if nargin < 2, m = n; end;

% gamma = parameter for strengthened SPR
if nargin < 3, gamma = 0; end;

% Quadratic cone of dimension >=3 for SeDuMi
% So if m<2, let m=2 and cancel leading coefficients in C
q = max(2-m,0); m = m+q;

% Build matrices for SeDuMi with decision variables:
% R,c1,..cm: controller parameters in a quadratic cone
% P1,P2,..PN: vertex SDP matrices of size p+1
p = max(n,m);
A = sparse(1+m+N*(p+1)^2,N*(p+1)+q);
b = sparse(size(A,2),1);
K.q = m+1;
K.s = (p+1)*ones(1,N);

% Toeplitz projection matrix
P = sparse((p+1)^2,(p+1));
for d = 0:p,
  v = diag(ones(1,p+1-d),d)+diag(ones(1,p+1-d),-d);
  P(:,d+1) = -v(:)/2;
end;

for k = 1:N,

  % Build Jury matrix
```

```
  coef = zeros(1,p+1);
  coef(1:n+1) = Ac(k,:);
  JuryD = flipud(hankel(fliplr(coef)));
  JuryU = hankel(coef);

  % Product c'*ak+ak'*c-2*gamma*ak'*ak
  cols = 1+(k-1)*(p+1):k*(p+1);
  A(2:m+1, cols) = JuryD(2:m+1,:)+JuryU(2:m+1,:);
  b(cols) = -JuryD(1,:)-JuryU(1,:)+2*gamma*(JuryU*JuryD(:,1))';

  % Projection on Pk
  A(2+m+(k-1)*(p+1)^2:1+m+k*(p+1)^2, cols) = P;

end;

% When necessary, cancel leading coefficients in controller
for d = 1:q,
 A(m+2-d,N*(p+1)+d) = 1;
end;

% Call SeDuMi
pars.numtol = 1e-3;
[x,y,info] = sedumi(A,b,[],K,pars);
if any([info.numerr info.pinf info.dinf]),
  error('No controller was found.');
end;

% Retrieve controller coefficients
c = [1 x(2:m+1)'];

% End of file ROBUSPR.M
```

References

[Alkire, 2001] B. Alkire and L. Vandenberghe "Convex Optimization Problems Involving Finite Autocorrelation Sequences", *Research Report*, Electrical Engineering Department, University of California, Los Angeles, January 2001.

[Anderson, 1990] B. D. O. Anderson, S. Dasgupta, P. Khargonekar, F. J. Kraus and M. Mansour "Robust Strict Positive Realness: Characterization and Construction", *IEEE Transactions on Circuits and Systems*, Vol. 37, No. 7, pp. 869–876, 1990.

[Anderson, 1994] B. D. O. Anderson and I. D. Landau "Least Squares Identification and the Robust Strict Positive Real Property", *IEEE Transactions on Circuits and Systems*, Part I, Vol. 41, No. 9, pp. 601–607, 1994.

[Dasgupta, 1987] S. Dasgupta and A. S. Bhagwat "Conditions for Designing Strictly Positive Real Transfer Functions for Adaptive Output Error Identification", *IEEE Transactions on Circuits and Systems*, Vol. 34, pp. 731–736, 1987.

[Henrion, 2000] D. Henrion "Linear Matrix Inequalities for Robust Strictly Positive Real Design", *Research Report*, No. 00571, LAAS-CNRS, Toulouse, France, December 2000.

[Jury, 1974] E. I. Jury "Inners and Stability of Dynamic Systems", *Wiley*, New York, 1974.

[Genin, 2000a] Y. Genin, Y. Nesterov and P. Van Dooren "Optimization over Positive Polynomial Matrices", *Proceedings of the International Symposium on Mathematical Theory of Networks and Systems*, Perpignan, France, June 2000.

[Genin, 2000b] Y. Genin, Y. Hachez, Y. Nesterov and P. Van Dooren "Convex Optimization over Positive Polynomials and Filter Design", *Proceedings of the IEE Control Conference*, Cambridge, England, September 2000.

[Kučera, 1979] V. Kučera "Discrete Linear Control: The Polynomial Approach", *John Wiley and Sons*, Chichester, 1979.

[Lasserre, 2001] J. B. Lasserre "Global Optimization with Polynomials and the Problem of Moments", *SIAM Journal on Optimization*, Vol. 11, No. 3, pp. 796–817, 2001.

[Mosquera, 1999] C. Mosquera, L. Leyra, F. Pérez "The Robust SPR Problem: Design Algorithms and New Applications", *Signal Processing*, Vol. 76, pp. 269–283, 1999.

[Mosquera, 2001] C. Mosquera, F. Pérez "Algebraic Solution to the Robust SPR Problem for Two Polynomials", *Automatica*, Vol. 37, No. 4, pp. 757–762, 2001.

[Nesterov, 2000] Y. Nesterov "Squared Functional Systems and Optimization Problems", Chapter 17, pp. 405–440 in H. Frenk, K. Roos, T. Terlaky and S. Zhang (Editors) "High Performance Optimization", Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[Parrilo, 2000] P. A. Parrilo "Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization", *PhD Thesis*, California Institute of Technology, Pasadena, California, 2000.

[Polyx Ltd., 2001] Polyx, Ltd. "The Polynomial Toolbox for Matlab", Version 2.5 released in 2001. Version 3.0 to be released soon. See `www.polyx.cz`.

[Šiljak, 1998] D. D. Šiljak and M. D. Šiljak "Nonnegativity of Uncertain Polynomials", *Mathematical Problems in Engineering*, Vol. 4, pp. 135–163, 1998.

[Sturm, 1999] J. F. Sturm "Using SeDuMi 1.02, a Matlab Toolbox for Optimization over Symmetric Cones", *Optimization Methods and Software*, Vol. 11-12, pp. 625–653, 1999. See also `fewcal.kub.nl/sturm`.

[Vandenberghe, 1996] L. Vandenberghe and S. P. Boyd "Semidefinite Programming", *SIAM Review*, Vol. 38, No. 1, pp. 49–95, 1996.