SPRST - A ROBUST STRICTLY POSITIVE REAL SYNTHESIS TOOLBOX FOR MATLAB [1]

Qiang Guan * **Wensheng Yu** *,*** **Long Wang** **

* *Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, 100080, P. R. China.*
** *Center for Systems and Control, Department of Mechanics and Engineering Science, Peking University, Beijing, 1000871, P. R. China.*
*** *National Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing, 100084, P. R. China.*

Abstract: A new robust Strictly Positive Real Synthesis Toolbox (SPRSt) for use with *Matlab* has been developed. Some algorithms for robust Strictly Positive Real Synthesis are introduced briefly and the use of the toolbox is illustrated by several examples. At last a link to downloadable code is provided.

Keywords: *Matlab* Toolbox, Computer-aided system design, Strict Positive Realness (SPR), Weak Strict Positive Realness (WSPR), Robustness.

1. INTRODUCTION

The strict positive realness (SPR) of a transfer function is important performance specification and plays a critical role in various fields such as absolute stability/hyperstabilty theory (Popov, 1973), passive analysis (Desoer and Vidyasagar, 1975), quadratic optimal control (Anderson and Moore, 1970) and adaptive system theory (Landau, 1979). In recent years, motivated by the parametrization approach in the robust stability analysis (Bhattacharrya *et al.*, 1995; Barmish, 1994; Huang, 2003), much attention has been paid to the study of robust positive realness of dynamic systems, and much progress has been made. Dasgupta and Bhagwat first addressed the SPR problem of interval systems (Dasgupta and Bhagwat, 1987). It was proved by Chapellat et.al. (Chapellat *et al.*, 1991), Wang and Huang (Wang and Huang, 1991) that the strict positive realness of an entire family of interval transfer functions can be ascertained by the same property of prescribed eight vertex transfer functions. Meanwhile, much progress on the robust strictly positive real synthesis has been made during the past decades.

The basic statement of the robust strictly positive real synthesis is as follows: Given an *n-th* order robustly stable polynomial set F, does there exist, and how can we construct a (fixed) polynomial $b(s)$ such that, $\forall a(s) \in F$, $b(s)/a(s)$ is strict positive realness?

For the robust strictly positive realness synthesis problem above, existing results show that: If the entries of F have the same even (or odd) parts, such a polynomial $b(s)$ always exits (Hollot *et al.*, 1989; Huang *et al.*, 1990; Patel and Datta, 1997); If F is a lower order ($n \leq 4$) stable interval polynomial set, such a polynomial $b(s)$ always exists (Anderson *et al.*, 1990; Hollot *et al.*, 1989; Huang *et al.*, 1990; Marquez and Agathoklis, 1998; Wang and Yu, 1999; Wang and Yu, 2000; Wang and

[1] This work is supported by National Natural Science Foundation of China (No.60204006, No. 69925307 and No. 60334020), National Key Basic Research and Development Program (No. 2002CB312200) and National Laboratory of Intelligent Technology and Systems of Tsinghua University.

Yu, 2001; Yu, 1998; Yu and Wang, 2001*a*); If F is a stable polynomial segment, such a polynomial *b(s)* always exists (Wang and Yu, 1999; Wang and Yu, 2000; Wang and Yu, 2001; Yu and Huang, 1998; Yu and Wang, 2001*b*; Yu and Wang, 2001*c*; Yu and Wang, 2003; Yu *et al.*, 2003*a*; Yu *et al.*, 2004); Some sufficient conditions for robust synthesis are presented (Anderson *et al.*, 1990; Bester and Zeheb, 1993; Dasgupta and Bhagwat, 1987; Marquez and Agathoklis, 1998; Wang and Yu, 1999; Wang and Yu, 2000; Yu, 1998). Especially, the design method proposed by Wang and Yu (Wang and Yu, 1999; Wang and Yu, 2000), based on the concept of weak strict positive realness (WSPR) and complete characterization of the SPR (WSPR) regions for transfer functions in coefficients space, is numerically efficient for high order systems and the derived conditions are necessary and sufficient conditions for stable segment polynomials and lower order stable interval polynomials ($n \leq 4$).

Furthermore, from Wang and Yu et. al. (Wang and Yu, 1999; Wang and Yu, 2000; Wang and Yu, 2001; Yu, 1998; Yu and Wang, 2001*b*; Yu and Wang, 2001*c*; Yu and Wang, 2003; Yu *et al.*, 2003*a*; Yu *et al.*, 2003*b*; Yu *et al.*, 2004), a design method for SPR synthesis called the geometric algorithm with order reduction is provided (Xie *et al.*, 2002*a*). It is a convex programming algorithm and computationally efficient for polynomial sets like segments, intervals and polytopes.

On one hand, there are other methods for SPR synthesis problem. For example, Bester and Zeheb (Bester and Zeheb, 1993) and Yu (Yu, 1998) deal with this problem using Matrix Equations (MEs) and Linear Matrix Inequalities (LMIs). However, the order of involved MEs or LMIs may be high, many variables must be introduced, and there is no theoretic result of the feasible conditions for the MEs or LMIs. On the other hand, Xie et. al. (Xie *et al.*, 2002*b*) has used Genetic Algorithm (GA) in SPR synthesis.

In this paper, we present a toolbox for *Matlab* integrated with these algorithms, that allows the user to solve SPR synthesis problem with very little effort.

The remainder of this paper is organized as follows. In Section II, preliminaries and our latest progress are introduced. Section III deals with algorithms which are implemented in this toolbox, especially for the geometric algorithm with order reduction. The usage of this toolbox is presented in Section IV. Numerical examples are provided to show the efficiency of this toolbox in Section V. At last, a link to downloadable toolbox and its code is available.

2. PRELIMINARIES

In this paper, P^n stands for the set of *n-th* order polynomials of s with real coefficients, R stands for the field of real numbers, R^n stands for n-dimensional real field, $H^n \subset P^n$ stands for the set of *n-th* order Hurwitz stable polynomials and $\partial(P)$ stands for the order of polynomial $P(\cdot)$.

In the following definitions (Wang and Yu, 1999; Wang and Yu, 2000), $b(\cdot) \in P^n$, $a(\cdot) \in P^m$, $p(s) = b(s)/a(s)$ is a rational function.

Definition 1. $p(s)$ is said to be strictly positive real (SPR), denote as $p(s) \in SPR$, if $b(s) \in P^n$, $a(s) \in H^n$, and $\forall\omega \in R$, Re$[p(j\omega)] > 0$.

Definition 2. $p(s)$ is said to be weak strictly positive real (WSPR), denote as $p(s) \in WSPR$, if $b(s) \in P^{n-1}$, $a(s) \in H^n$, and $\forall\omega \in R$, Re$[p(j\omega)] > 0$.

Definition 3. Given $a(s) \in H^n$, the set of coefficients (in R^{n+1}) of all the $b(s)$'s in P^n such that $p(s) := \dfrac{b(s)}{a(s)} \in SPR$ is said to be the SPR region associated with $a(s)$, denote as Ω_a.

Definition 4. Given $a(s) \in H^n$, the set of coefficients (in R^n) of all the $b(s)$'s in P^{n-1} such that $p(s) := \dfrac{b(s)}{a(s)} \in WSPR$ is said to be the WSPR region associated with $a(s)$, denote as Ω_a^W.

For notational convenience, Ω_a (Ω_a^W) sometimes also stands for the set of all the polynomials $b(s)$ in P^n (P^{n-1}), such that $p(s) := \dfrac{b(s)}{a(s)} \in SPR(WSPR)$.

Without loss of generality, let $a(s) = s^n + a_1 s^{n-1} + \cdots + a_n \in H^n$, denote Ω_{1a} as the set of the coefficients of all the $b(s) = s^n + x_1 s^{n-1} + \cdots + x_n \in P^n$, i.e., $(x_1, x_2, \cdots, x_n)$ in R^n such that $p(s) := \dfrac{b(s)}{a(s)} \in SPR$; and denote Ω_{1a}^w as the set of the coefficients of all the $b(s) = s^{n-1} + x_2 s^{n-2} + \cdots + x_n \in P^{n-1}$, i.e., $(x_2, x_3, \cdots, x_n)$ in R^{n-1} such that $p(s) := \dfrac{b(s)}{a(s)} \in WSPR$.

From Wang and Yu (Wang and Yu, 2000), we know that the boundary of every entry of b is: $(x_2, x_3, \cdots, x_n) \in \Omega_{1a}^W$, $\Omega_{1a}^W \subset \{(x_2, x_3, \cdots, x_n) | 0 < x_2 \leq a_1, \cdots, 0 < x_n < a_{n-1}\}$.

Property 1. (Wang and Yu, 2000) Given $a(s) \in H^n$, if $(x_2, x_3, \cdots, x_n) \in \Omega_{1a}^W$, then $\forall (1, \alpha_1, \alpha_2, \cdots, \alpha_n) \in R^{n+1}$, we cam take sufficient small $\varepsilon > 0$ such that $(0, 1, x_2, x_3, \cdots, x_n) + \varepsilon(1, \alpha_1, \alpha_2, \cdots, \alpha_n) \in \Omega_a$.

Since Ω_a and Ω_{1a} are both unbounded sets (Hollot *et al.*, 1989; Wang and Yu, 2000), when considering the SPR synthesis problem, it is hardly tractable operating on unbounded set to check the intersection of SPR regions. On the other hand, from Wang and Yu (Wang and Yu, 1999; Wang and Yu, 2000), we can construct the finite search space for this problem. Thereby we first consider the WSPR problem. Furthermore, **Property 1** reveals the relationship between Ω_{1a}^W and Ω_a and plays an important role in robust SPR synthesis.

In what follows, we first introduce some notations, which are necessary in discussion below. Let

$$a(s)=s^n+a_1s^{n-1}+\cdots+a_n\in H^n, \quad (1)$$

$$b(s)=x_0s^n+x_1s^{n-1}+\cdots+x_n\in P^n, \quad (2)$$

Then $\forall\omega \in R$, we have

$$\begin{aligned} Re[\tfrac{b(j\omega)}{a(j\omega)}] &= \tfrac{1}{|a(j\omega)|^2} Re[b(j\omega)a(-j\omega)] \\ &= \tfrac{1}{|a(j\omega)|^2}\sum_{l=0}^{n} c_l\omega^{2(n-l)} \end{aligned}$$

where $c_l = \displaystyle\sum_{k=0}^{n} a_k x_{2l-k}(-1)^{l+k}$, $a_0 = 1$ and if $i<0$ or $i>n$, let $a_i=0, x_i=0, l=0,1,2,\cdots,n$, when considering the WSPR problem, we take $x_0=0$.

Introducing the Matrices:

$$H_a := \begin{bmatrix} a_1 & 1 & 0 & 0 & 0 & \cdots & 0 \\ a_3 & a_2 & a_1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ a_{2n-1} & a_{2n-2} & a_{2n-3} & a_{2n-4} & a_{2n-5} & \cdots & a_n \end{bmatrix}$$

$$E_a := \begin{bmatrix} 1 & & & & \\ & -1 & & & \\ & & 1 & & \\ & & & -1 & \\ & & & & \ddots \end{bmatrix}$$

$$A:=E_aH_aE_a \qquad b:=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad c:=\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

where $a_i = 0$ when $i > n$, H_a is the Hurwitz matrix of $a(s)$. Then it is easy to verify that

$$c=Ab \quad (3)$$

Since $\dfrac{b(s)}{a(s)} \in WSPR \Leftrightarrow \displaystyle\sum_{l=1}^{n} c_l\omega^{2(n-l)} > 0$, in order to simplify the WSPR synthesis problem, attention has been focused on the vector c. Here, denote $f(\omega^2) = \displaystyle\sum_{l=1}^{n} c_l\omega^{2(n-l)}$.

The following is the important results we have achieved recently.

Lemma 1. (Yu *et al.*, 2003*a*; Yu *et al.*, 2004) Suppose $a(s) = s^n + a_1s^{n-1} + \cdots + a_n \in H^n$, then for every $k \in \{1,2,\cdots,n-2\}$, the following quadratic curve is an ellipse in the first quadrant (i.e., $x_i \geq 0, i = 1,2,\cdots,n-1$) of the R^{n-1} space $(x_1,x_2,\cdots,x_{n-1})$:

$$\begin{cases} c_{k+1}^2 - 4c_kc_{k+2} = 0 \\ c_l = 0 \\ l \in [1,2,\cdots,n], l \neq k,k+1,k+2 \end{cases}$$

and the ellipse is tangent with the line

$$\begin{cases} c_l = 0 \\ l \in [1,2,\cdots,n], l \neq k+1,k+2 \end{cases}$$

and the line

$$\begin{cases} c_l = 0 \\ l \in [1,2,\cdots,n], l \neq k,k+1 \end{cases}$$

respectively, where $c_l = \displaystyle\sum_{j=0}^{n} a_j x_{2l-j-1}(-1)^{l+j}$, $l = 1,2,\cdots,n$, $a_0 = 1, x_0 = 1, a_i = 0$ if $i > n$ or $i < 0$ and $x_i = 0$ if $i < 0$ or $i > n-1$.

For notational simplicity, for $a(s) = s^n + a_1s^{n-1} + \cdots + a_n \in H^n$, $\forall k \in \{1,2,\cdots,n-2\}$, denote

$$\begin{aligned} \Omega_{ek}^a := \{ & (x_1,x_2,\cdots,x_{n-1}) | c_{k+1}^2 - 4c_kc_{k+2} < 0, \\ & c_l = 0, l \in \{1,2,\cdots,n\}, l \neq k,k+1,k+2\} \end{aligned}$$

where $c_l = \displaystyle\sum_{j=0}^{n} a_j x_{2l-j-1}(-1)^{l+j}$, $l = 1,2,\cdots,n$, $a_0 = 1, x_0 = 1, a_i = 0$ if $i > n$ or $i < 0$ and $x_i = 0$ if $i < 0$ or $i > n-1$.

In what follows, (A,B) stands for the set of points in the line segment connecting the point A and the point B in the the R^{n-1} space $(x_1,x_2,\cdots,x_{n-1})$, not including the end point A and B. Denote

$$\begin{aligned} \Omega^a := \{ & (x_1,x_2,\cdots,x_{n-1}) | \\ & (x_1,x_2,\cdots,x_{n-1}) \in \cup_{i=1,i<j\leq n-2}^{n-3} (A_i,A_j), \\ & \forall A_i \in \Omega_{ei}^a, i \in \{1,2,\cdots,n-2\}\}. \end{aligned}$$

Lemma 2. (Yu *et al.*, 2003*a*; Yu *et al.*, 2004) Suppose $a(s) = s^n + a_1s^{n-1} + \cdots + a_n \in H^n$, take $(x_1,x_2,\cdots,x_{n-1}) \in \Omega^a$, and let $c(s) = s^{n-1} + (x_1 - \varepsilon)s^{n-2} + x_2s^{n-3} + \cdots + x_{n-2}s + (x_{n-1} + \varepsilon)$ (ε is a sufficient small positive number), then for $\dfrac{c(s)}{a(s)}$ we have $\forall\omega \in R, Re[\dfrac{c(j\omega)}{a(j\omega)}] > 0$, viz. $(x_1 - \varepsilon, x_2, \cdots, x_{n-2}, x_{n-1} + \varepsilon) \in \Omega_{1a}^W$.

By Yu et. al. (Yu *et al.*, 2003*a*; Yu *et al.*, 2004), we have:

Theorem 1. (Yu *et al.*, 2003*a*; Yu *et al.*, 2004) If $F = \{a_i(s) = s^n + \sum_{l=1}^{n} a_l^{(i)} s^{n-l}, i = 1,2\}$ is the set of two endpoint polynomials of a stable segment of polynomials (convex combination), then we have $\cap_{i=1}^{2}\Omega_{1a_i}^W \neq \phi$

Meanwhile, From Wang and Yu (Wang and Yu, 2001; Yu and Wang, 2001*a*; Yu and Wang, 2001*b*), we have:

Theorem 2. (Wang and Yu, 2001; Yu and Wang, 2001*a*; Yu and Wang, 2001*b*) If $F = \{a_i(s) = s^n + \sum_{l=1}^{n} a_l^{(i)} s^{n-l}, i = 1,2,3,4\}$ is the set of four Kharitonov vertex polynomials of a lower order ($n \leq 4$) stable interval of polynomials family, then we have $\cap_{i=1}^{4}\Omega_{1a_i}^W \neq \phi$

Combining Lemma 1, Lemma 2, Theorem 1, Theorem 2 and Property 1, thus we can achieve the following important result:

If F is a polynomial segment or a lower order ($n \leq 4$) interval polynomial set , then the existence of a polynomial $b(s)$ such that $\forall a(s) \in F$, $b(s)/a(s)$ is strict positive realness is equivalent to that F is robustly stable.

In fact, the main idea of the geometric algorithm with order reduction originates from the results above.

3. ALGORITHMS FOR SPR SYNTHESIS

3.1 The Geometric Order-reduced Algorithm

For a general polynomial family F with n vertices $a^k(s)$, k=$1, 2, \cdots, m$, denote $upper(x_i) = min(a^1_{i-1}, a^2_{i-1}, \cdots, a^m_{i-1})$, where a^k_{i-1} represents the *(i-1)th* entry of the a^k, $k = 1, 2, \cdots, m$ (Wang and Yu, 2000). For the Eq.(3) $c = Ab$, let $x_1 = 1$. Suppose there are only three continuous non-zero entries in vector c , i.e. $c^T = [0, \cdots, 0, c_i, c_{i+1}, c_{i+2}, 0, \cdots, 0]$, $i = 1, 2, \cdots, n-2$.
From Eq.(3), we can get

$$f(t) = (c_i t^2 + c_{i+1} t + c_{i+2}) \cdot t^{n-i-2}$$

since c_i is a linear combination of x_i, it is easy to see that there are only two entries of vector b that are uncertain, denote as x_{j1} and x_{j2}.
Let $f(t) > 0, t \in [0, +\infty)$, for this purpose, consider
$\triangle(x_{j1}, x_{j2}) = c^2_{i+1} - 4c_{i+2}c_i < 0, c_{i+2} > 0, c_i > 0$
According to Lemma 1, we can guarantee that $\triangle(x_{j1}, x_{j2})$ be an ellipse. Therefore, denote B as a 3×3 dimension symmetric matrix in following Eq.(4). Thus the WSPR synthesis problem can be transformed to the feasible problem of the following quadratic inequalities:

$$\begin{gathered}\triangle(x_{j1},x_{j2})=[1 \;\; x_{j1} \;\; x_{j2}]\cdot B\cdot \begin{bmatrix}1\\ x_{j1}\\ x_{j2}\end{bmatrix}\\ 0<x_{j1}<upper(x_{j1})\\ 0<x_{j2}<upper(x_{j2}). \qquad (4)\end{gathered}$$

It is rather easy to solve, and in literature many efficient methods cam wok it well, in our toolbox, we use gridding and testing.
Based on the above discussion, the main procedures of the geometric algorithm are summarized as follows: (Xie *et al.*, 2002*a*)

Step 1: For the input vertices of polynomials, test the robust stability of convex hull of F (involving m vertices), i.e. $\overline{F}$, if $\overline{F}$ is robustly stable, then go to step 2; otherwise print "there does not exist such a *b(s)*". (by Definition 1 and 2)

Step 2: Choose a vertex polynomial $a^k(s)$ from F, k=$1, 2, \cdots, m$. Set c^T=$[0, \cdots, 0, c_i, c_{i+1}, c_{i+2}, 0, \cdots, 0]$, i=$1, 2, \cdots, n-2$. Solve the Eq.(3) and yield the vector b with 2 variables. Search feasible solutions of the Eq.(4), select a sufficient small real $\varepsilon > 0$, thus obtaining b^1 (by Lemma 2). Test whether this solution b^1 belong to $\cap^m_{k=1}\Omega^W_{1a^k}$, If yes, go to step 5, else go to step 3.

Step 3: $i = i + 1$; if $i > n - 2$, go to step 4, else go to step 5.

Step 4: Change a^k with another polynomial that has not be chosen. Go to step 2.

Step 5: Take a sufficiently small $\varepsilon_1 > 0$ such that $(\varepsilon_1, x_1, x_2, \cdots, x_n) \in \cap^m_{k=1}\Omega_{a^k}$. Then the *n-th* order polynomial $b(s)$ with coefficients $(\varepsilon_1, x_1, x_2, \cdots, x_n)$ satisfies the design requirement(by Property 1). The complete discrimination system for polynomials (Yang *et al.*, 1996) has been applied to test whether the solution is required.

3.2 The Algorithm based on Genetic Algorithm

As a general optimization problem solver, due to its intrinsic parallelism and some intelligent properties, GA has been applied successfully to problems where heuristic solution are not available or generally lead to unsatisfactory results. An algorithm based on GA for the robust SPR synthesis is well discussed in (Xie *et al.*, 2002*b*). More details and procedures can be found in (Xie *et al.*, 2002*b*).

3.3 The Algorithm based on LMIs

It is well known that many problems in systems and control can be formulated as "Linear Matrix Inequalities" (LMIs) problems. The algorithm based on LMIs which is implemented in our toolbox stems from much work involving applying LMIs method to robust SPR synthesis (Bester and Zeheb, 1993; Yu, 1998). Unlike the geometric algorithm with order reduction and the algorithm based on GA, it deals with state-space model. Its main idea is that we can transform the SPR (WSPR) problem to the LMIs problems with constraints using Positive-Real Lemma (Popov, 1973; Bhattacharrya *et al.*, 1995) and some pertinent results (Bester and Zeheb, 1993; Yu, 1998). Thus we can take advantage of LMI Toolbox available in *Matlab* to solve it. It should be admitted that by introducing the concept of weak strict positive realness (WSPR), the algorithm based on LMIs reduces computational burden.

4. INFORMATION ABOUT THE TOOLBOX

In order to help users to solve the SPR synthesis problem efficiently, we have developed a complete package (toolbox) for *Matlab* we called SPRSt, that allows the user to solve the SPR synthesis problem using algorithms which are introduced briefly in Section III. With it, users can get the results of their problems online without having to write complicated code or really even understand much about these algorithms.

Like many other toolboxes, the SPRSt is composed of many functions (M files), including some main solvers and other support functions. The Table 1 shows general information about it.
The usage of some main solver are as follows:

Table 1. General information for SPRSt

GUI	
sprdemo	- Main function for GUI
Genetic	Algorithm
sprgene	- Using GA in SPR synthesis
sprfitc	- Computing the fitness
genemain	- Standard genetic algorithm function
wsprgene	- Using GA in WSPR synthesis
Geometry	Algorithm
spr_reduct	- Main function
wspr_inter	- compute the intersection of WSPR
	- regions using order_reduced
	- in the ellipse domain
LMIs	Algorithm
sprlmi	- Using LMIs in SPR synthesis
wsprlmi	- Using LMIs in WSPR synthesis
Complete	Polynomials Discrimination System
comtest	- Determine whether the polynomials
	- set a and polynomial b form SPR
	- or WSPR
compoly	- Using the Complete Discrimination
	- System for Polynomials to determent
	- whether the polynomial has real root
Miscellaneous	functions
monic	- Monic the polynomial set matrix
hurwitz	- Verifying the H-Stability
sabsolue	- Get the square of a polynomial
	- in jω plane
wspr2spr	- Find SPR solution using WSPR

(1) Function spr_reduct
This function solves the SPR synthesis problem using the geometric algorithm with order reduction.
Syntax:
$[r, b] = spr\_reduct(a, step)$
Input:
- a: the matrix stands for the input vertices of polynomials, and each row of this matrix represents a polynomial.
- step: the size of grid. When searching the feasible solution of quadratic inequalities in an ellipse (please see Section III), gridding method is used. The default value is 50.

output:
- r: the result returned by the geometric algorithm.
  1 - successfully find the vector b satisfies the requirement of SPR.
  0 - fail to find the vector b meets the need of SPR.
- b: the vector stands for a *n-th* polynomial.
  when r=1, it is the solution vector of SPR.
  when r=0, every entry of it is zero.

(2) Function sprgene
This function solves the SPR synthesis problem based on genetic algorithm. It calls function wsprgene whose purpose is the WSPR problem.
Syntax:
$[r, b] = sprgene(a, ValMulti, options)$
Input:
- ValMulti: the multiple of the upper of coefficients of the search space which is used in WSPR synthesis.
- options: the vector holds some basic arguments needed by standard GA .

(3) Function sprlmi
This function solves the SPR synthesis problem based on LMIs.
Syntax:
$[r, b] = sprlmi(a)$

(4) Function wsprlmi
This function solves the WSPR synthesis problem based on LMIs.
Syntax:
$[r, b] = wsprlmi(a)$

For some conveniences, besides some functions, a graphic user interface (GUI) demo program is provided in SPRSt to show how to use these functions.

5. EXAMPLES

In this section, a vector form $[1\ a_1\ a_2\ \cdots\ a_n]$ represents the polynomial $a(s) = s^n + a_1 s^{n-1} + \cdots + a_n$. Each row of a matrix stands for a polynomial when there are many polynomials in question.

Example 1. Consider a family of 6-th order polynomial set (segment) F:

$$\begin{bmatrix} 1 & 12 & 70 & 300 & 500 & 600 & 300 \\ 1 & 14 & 60 & 280 & 490 & 650 & 400 \end{bmatrix}$$

It is easy to see that the convex hull $\overline{F}$ is robustly Hurwitz stable. To solve this SPR synthesis problem with the SPRSt, the solver function spr_reduct can be invoked and the following result can be yielded.
the output:

```
find a spr polynomial:
1 123 147.1 600.5 588.2 1213.2 1.2
```

Example 2. Consider a family of 4-th order polynomial set (interval) F:

$$\begin{bmatrix} 1 & 89 & 56 & 88 & 1 \\ 1 & 11 & 56 & 88 & 50 \\ 1 & 89 & 56 & 88 & 50 \\ 1 & 11 & 56 & 88 & 1 \end{bmatrix}$$

It is easy to see that the convex hull $\overline{F}$ is robustly Hurwitz stable. Invoking the solver function sprgene to solve this SPR synthesis problem, yields the following result:
the output:

```
find a spr polynomial b(s):
1     9.6947     234.6732     150.4671     4.4695
```

Example 3. Consider a family of 3-th order polynomial set (segment) F:

$$\begin{bmatrix} 1 & 1.71 & 5.39 & 0.47 \\ 1 & 8.1 & 4 & 8 \end{bmatrix}$$

It is easy to see that the convex hull $\overline{F}$ is robustly Hurwitz stable. Invoking the solver function sprlmi yields the following result:
the output:

```
find a spr polynomial b(s):
1.0000     7.6528     6.9173     5.7821
```

Example 4. Consider a family of 9-th order polynomial set (segment) F:

$$\begin{bmatrix} 1 & 11 & 52 & 145 & 266 & 331 & 280 & 155 & 49 & 6 \\ 1 & 11 & 52 & 146 & 265.5 & 332 & 278.5 & 151 & 48 & 2 \end{bmatrix}$$

It is easy to see that the convex hull $\overline{F}$ is robustly Hurwitz stable. Invoking the solver function wsprlmi yields the following result:

the output:

```
find a wspr polynomial b(s):
1.0000      4.6310     15.7490     29.5778
37.3166   31.0205     16.1880      4.4975
0.4651
```

6. CONCLUSIONS

The suite of functions and programs included with the SPR Synthesis Toolbox are useful both in researching strict positive realness and applying SPR to control systems. The algorithms implemented here are efficient and works continues on improvement. As it stands, the toolbox allows the researcher/engineer to solve the SPR synthesis problem without having to write custom code from the ground up. The included comments make the suite easily modified to fit more specific requirements. The package and its code can be downloaded through *http://www.ia.ac.cn/personal/wensheng.yu/sprst.zip*. Please send email to *qiang.guan@mail.ia.ac.cn* if you wish to be informed about future update this software.

REFERENCES

Anderson, B. D. O. and J. B. Moore (1970). *Linear Optimal Control*. Prentice Hall. New York.

Anderson, B. D. O., S. Dasgupta and P. Khargonekar (1990). Robust strict positive realness: characterization and construction. *IEEE Trans. Circuits and Systems* **CAS-37**, 869–876.

Barmish, B. R. (1994). *New Tools for Robustness of Linear Systems*. MacMillan Publishing Company. New York.

Bester, A. and E. Zeheb (1993). Design of robust strictly positive realness transfer functions. *IEEE Trans. Circuits and Systems* **CAS-40**, 573–580.

Bhattacharryya, S. P., H. Chapellat and L. H. Keel (1995). *Robust Control: The Parametric Approach*. Prentice Hall. New York.

Chapellat, H., M. Dahleh and S. P. Bhattacharyya (1991). On robust nonlinear stability of interval control systems. *IEEE Trans. Automat. Contr.* **AC-36**, 59–69.

Dasgupta, S. and A. S. Bhagwat (1987). Conditions for designing strictly positive real transfer functions for adaptive output error identification. *IEEE Trans. Circuit and Systems* **CAS-34**, 731–736.

Desoer, C. A. and M. Vidyasagar (1975). *Feedback systems: Input-Output Properties*. Academic Press. San Diego.

Hollot, C.V., L. Huang and Z. Xu (1989). Designing strictly positive real transfer function families: A necessary and sufficient condition for low degree and structured families. In: *Proc. of International Conference on Mathematical Theory of Network and Systems*. pp. 215–227.

Huang, L. (2003). *The Theoretic Fundament for Stability and Robustness*. Science Press. Beijing.

Huang, L., C.V. Hollot and Z. Xu (1990). Robust analysis of strictly positive real function set. In: *Preprint of the second Japan-China Joint Symposium on System Control Theory and its Applications*. pp. 210–220.

Landau, I. D. (1979). *Adaptive control: the Model Reference Approach*. Marcel Deker. New York.

Marquez, H. J. and P. Agathoklis (1998). On the existence of robust strictly positive real rational functions. *IEEE Trans. Circuits and Systems* **CAS-45**, 962–967.

Patel, V. V. and K. B. Datta (1997). Classification of units in $H\infty$ and an alternative proof of khartionov's theorem. *IEEE Trans. Circuits and Systems* **CAS-44**, 454–458.

Popov, V. M. (1973). *Hyperstability of control systems*. Springer-Verlag. New York.

Wang, L. and L. Huang (1991). Finite verification of strict positive realness of interval transfer function. *Chinese Science Bulletin* **36**, 262–268.

Wang, L. and W. Yu (1999). A new approach to robust synthesis of strictly positive real transfer functions. *Stability and Control: Theory and Applications* **2**, 13–24.

Wang, L. and W. Yu (2000). Complete characterization of strict positive realness regions and robust strictly positive real synthesis method. *Science in China* **E-43**, 97–112.

Wang, L. and W. Yu (2001). Robust SPR synthesis for lower-order polynomial segments and interval polynomials. In: *Proceedings of ACC 2001*. pp. 3612–3617.

Xie, L., L. Wang and W. Yu (2002*a*). A new geometric algorithm with order reduction for robust strictly positive real synthesis. In: *The 41st IEEE Conference on Decision and Control(CDC2002)*. Las Vegas, USA. pp. 1844–1849.

Xie, L., L. Wang, W. Yu and Y. Qiu (2002*b*). Robust strictly positive real synthesis based on genetic algorithm. In: *Proceedings of the 15th IFAC Triennial World Congress*. Barcelona, Spain.

Yang, L., X. R. Hou and Z. B. Zeng (1996). A complete discrimination system for polynomials. *Science in China* **E-39**, 628–646.

Yu, W. (1998). Robust SPR synthesis and robust stability analysis. PhD thesis. Peking University.

Yu, W. and L. Huang (1998). Necessay amd sufficient conditions for strictly positive real stablization of low-order systems. *Chinese Science Bulletin* **43**, 275–279.

Yu, W. and L. Wang (2001*a*). Anderson's claim on fourth-order convex SPR synthesis is true. *IEEE Trans. Circuits and Systems* **CAS-48**, 506–509.

Yu, W. and L. Wang (2001*b*). Robust SPR synthesis for fourth-order convex combinations. *Progress in Natural Science* **11**, 461–467.

Yu, W. and L. Wang (2001*c*). Robust strictly positive real synthesis for convex combinations of fifth-order polynomial. In: *Proceedings of the IEEE Symposium on Circuits and Systems Conference (ISCAS 2001)*. Vol. 1.1. Sydney, Australia. pp. 739–742.

Yu, W. and L. Wang (2003). Robust strictly positive real synthesis for convex combinations of sixth-order polynomials. In: *2003 American Control Conference (ACC 2003)*. pp. 3840–3845.

Yu, W., L. Wang and J. Ackermann (2003*a*). Solution to the general robust strictly positive real synthesis problem for polynomial segments. In: *2003 European Control Conference (ECC 2003)*. UK.

Yu, W., L. Wang and J. Ackermann (2004). Robust strictly positive real synthesis for polynomial families of arbitrary order. *Science in China* **F-47**, 475–489.

Yu, W., L. Wang and Y. Xiang (2003*b*). Robust strictly positive real synthesis of polynomial segments for discrete time systems. In: *The 42nd IEEE Conference on Decision and Control (CDC 2003)*. USA. pp. 622–627.