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/hyperstability 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 (Holotl 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; Holotl et al., 1989; Huang et al., 1990; Marquez and Agathoklis, 1998; Wang and Yu, 1999; Wang and Yu, 2000; Wang and
Yu, 2001; Yu, 1998; Yu and Wang, 2001a); 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, 2001b; Yu and Wang, 2001c; Yu and Wang, 2003; Yu et al., 2003a; 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, 2001b; Yu and Wang, 2001c; Yu and Wang, 2003; Yu et al., 2003a; Yu et al., 2003b; Yu et al., 2004), a design method for SPR synthesis called the geometric algorithm with order reduction is provided (Xie et al., 2002a). 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., 2002b) 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), $b(\cdot) \in P^n$, $a(\cdot) \in P^n$, $p(s) = b(s)/a(s)$ 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 $\forall \omega \in \mathbb{R}, \text{Re}\{p(j\omega)\} > 0$.

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

**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 $p(s) := \frac{b(s)}{a(s)} \in SPR$ is said to be the SPR region associated with $a(s)$, denote as $\Omega_a$.

**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 $p(s) := \frac{b(s)}{a(s)} \in WSPR$ is said to be the WSPR region associated with $a(s)$, denote as $\Omega^W_a$.

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

Without loss of generality, let $a(s) = s^n + a_1 s^{n-1} + \cdots + a_n \in H^n$, denote $\Omega_{1a}$ 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., $(x_1, x_2, \ldots, x_n) \in R^n$ such that $p(s) := \frac{b(s)}{a(s)} \in SPR$; and denote $\Omega^W_{1a}$ 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., $(x_2, x_3, \ldots, x_n) \in R^{n-1}$ such that $p(s) := \frac{b(s)}{a(s)} \in WSPR$.

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

**Property 1.** (Wang and Yu, 2000) Given $a(s) \in H^n$, if $(x_2, x_3, \ldots, x_n) \in \Omega^W_{1a}$ then $\forall (1, a_1, a_2, \ldots, a_n) \in R^{n+1}$, we can take sufficient small $\epsilon > 0$ such that $(0, 1, x_2, x_3, \ldots, x_n) + \epsilon(1, a_1, a_2, \ldots, a_n) \in \Omega_{1a}$.

Since $\Omega_{1a}$ and $\Omega^W_{1a}$ are both unbounded sets (Holot 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 \( \Omega^W_k \) and \( \Omega_\alpha \) and plays an important role in robust SPR synthesis.

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

\[
a(s) = a^n + a_1 a^{n-1} + \cdots + a_n \in H^n,
\]

\[
b(s) = b_0 s^n + b_1 s^{n-1} + \cdots + b_n \in H^n.
\]

Then \( \forall \omega \in R \), we have

\[
Re\left[ \frac{b(s)}{a(s)} \right] = \left| \frac{b(j\omega)}{a(j\omega)} e^{-j\omega T} \right| = \left| \frac{c(j\omega)}{a(j\omega)} \right|^2 = c_l \omega^{2(n-l)}
\]

where

\[
c_l = \sum_{k=0}^{n} a_k x_{2l-k} (-1)^{l+k}, \quad a_0 = 1 \quad \text{and if}
\]

\[i < 0 \quad \text{or} \quad i > n, \quad \text{let} \quad a_i = 0, \quad x_i = 0, \quad l = 0, 1, 2, \cdots, n,
\]

when considering the WSPR problem, we take \( x_0 = 0 \).

Introducing the Matrices:

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

\[
E_a := \begin{bmatrix}
  1 \\
  -1 \\
  1 \\
  \vdots \\
  \vdots \\
  \vdots \\
  -1
\end{bmatrix}
\]

\[
A := E_a H_a E_a
\]

\[
b := \begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{bmatrix}, \quad c := \begin{bmatrix}
  c_1 \\
  c_2 \\
  \vdots \\
  c_n
\end{bmatrix}
\]

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

\[
\frac{b(s)}{a(s)} \in WSPR \iff \sum_{l=1}^{n} c_l \omega^{2(n-l)} > 0,
\]

in order to simplify the WSPR synthesis problem, attention has been focused on the vector \( c \). Here, denote

\[
f(\omega^2) = \sum_{l=1}^{n} c_l \omega^{2(n-l)}.
\]

The following is the important results we have achieved recently.

**Lemma 1.** (Yu et al., 2003a; Yu et al., 2004) Suppose \( a(s) = s^n + a_1 s^{n-1} + \cdots + a_n \in H^n \), then for every \( k \in \{1, 2, \cdots, n-2\} \), the following quadratic curve is an ellipse in the first quadrant (i.e., \( x_i > 0, i = 1, 2, \cdots, n-1 \)) of the \( R^{n-1} \) space \( (x_1, x_2, \cdots, x_{n-1}) \):

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

and the ellipse is tangent with the line

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

and the line

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

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

For notational simplicity, for \( a(s) = s^n + a_1 s^{n-1} + \cdots + a_n \in H^n \), \( \forall k \in \{1, 2, \cdots, n-2\} \), denote

\[
\Omega_k := \{ (x_1, x_2, \cdots, x_{n-1}) | c_{l+1}^2 - 4c_{k}c_{k+l+2} < 0, \}
\]

\[
c_l = 0 \iff \{1, 2, \cdots, n, k \neq k+1, k+2 \}
\]

where \( c_l = \sum_{j=0}^{n} a_j x_{2l-j} (-1)^{l+j}, \ l = 1, 2, \cdots, n, \ a_0 = 1, x_0 = 1, a_i = 0 \) 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

\[
\Omega^a := \{ (x_1, x_2, \cdots, x_{n-1}) | c_{l+1}^2 - 4c_{k}c_{k+l+2} < 0, \}
\]

\[
c_l = 0 \iff \{1, 2, \cdots, n, k \neq k+1, k+2 \}
\]

\[
\forall \omega \in R, \ Re\left[ \frac{b(s)}{a(s)} \right] = c_l \omega^{2(n-l)} > 0, \ \text{viz.} \ (x_1 - c, x_2, \cdots, x_{n-2}, x_{n-1} + \epsilon) (\epsilon \ is \ a \ sufficient \ small \ positive \ number), \text{then for} \ \frac{b(s)}{a(s)} \ \text{we have}
\]

\[
\forall \omega \in R, \ Re\left[ \frac{b(s)}{a(s)} \right] > 0,
\]

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

**Theorem 1.** (Yu et al., 2003a; Yu et al., 2004) If

\[
F := \{ a(s) = s^n + \sum_{i=1}^{n} a_i s^{n-i}, i = 1, 2 \}
\]

is the set of two endpoint polynomials of a stable segment of polynomials (convex combination), then we have

\[
\forall \omega \in R, \ Re\left[ \frac{b(s)}{a(s)} \right] > 0,
\]

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

**Theorem 2.** (Wang and Yu, 2001; Yu and Wang, 2001a; Yu and Wang, 2001b) If

\[
F := \{ a(s) = s^n + \sum_{i=1}^{n} a_i s^{n-i}, i = 1, 2, 3, 4 \}
\]

is the set of four Kharitonov vertex polynomials of a lower order (\( n \leq 4 \)) stable interval of polynomials family, then we have

\[
\forall \omega \in R, \ Re\left[ \frac{b(s)}{a(s)} \right] > 0.
\]

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 \( a_k(s), k = 1, 2, \ldots, m \), denote upper \( (x_i) = \min(a_k^{i-1}, a_k^{i-1}, \ldots, a_k^{n-1}) \), where \( a_k^{i-1} \) represents the \((i-1)\)th entry of the \( a_k \), \( k = 1, 2, \ldots, m \) (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. \( c^T = [0, 0, c_i, c_i+1, c_i+2, 0, \ldots, 0] \), \( i = 1, 2, \ldots, n-2 \).

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
\[
\Delta(x_{j1}, x_{j2}) = c_{i+1}^2 - 4c_{i+2}c_i < 0, c_{i+2} > 0, c_i > 0
\]
According to Lemma 1, we can guarantee that \( \Delta(x_{j1}, x_{j2}) \) be an ellipse. Therefore, denote \( B \) as a \( 3 \times 3 \) dimension symmetric matrix in following Eq.(4). Thus the WSPR synthesis problem can be transformed to the feasible problem of the following quadratic inequalities:
\[
\begin{align*}
\Delta(x_{j1}, x_{j2}) &= |1 \times x_{j1} x_{j2}|B \begin{bmatrix} 1 \\ x_{j1} \\ x_{j2} \end{bmatrix} \\
&\leq 0 < c_{j1} < \text{upper}(x_{j1}) \\
&0 < c_{j2} < \text{upper}(x_{j2}).
\end{align*}
\]

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., 2002a)

Step 1: For the input vertices of polynomials, test the robust stability of convex hull of \( F \) (involving \( m \) vertices), i.e. \( \mathcal{F} \), if \( \mathcal{F} \) 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 \( F \), \( k = 1, 2, \ldots, m \). Set \( c^T = [0, 0, \ldots, 0, c_i, c_i+1, c_i+2, 0, \ldots, 0] \), \( i = 1, 2, \ldots, n-2 \). 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 \( b^l \) (by Lemma 2). Test whether this solution \( b^l \) belong to \( \cap_{k=1}^{m} \Omega_{b^l}^k \). 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 been chosen. Go to step 2.

Step 5: Take a sufficiently small \( \varepsilon_1 > 0 \) such that \( (\varepsilon_1, x_1, x_2, \ldots, x_n) \in \cap_{k=1}^{m} \Omega_{b^l}^k \). Then the \( n \)-th order polynomial \( b(s) \) with coefficients \( (\varepsilon_1, x_1, x_2, \ldots, x_n) \) 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., 2002b). More details and procedures can be found in (Xie et al., 2002b).

3.3 The Algorithm based on LMI

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; Bhattacharyya 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

<table>
<thead>
<tr>
<th>GUI</th>
<th>sprdemo</th>
<th>- Main function for GUI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genetic</td>
<td>Algorithm</td>
<td></td>
</tr>
<tr>
<td>sprgene</td>
<td>- Using GA in SPR synthesis</td>
<td></td>
</tr>
<tr>
<td>sprfitc</td>
<td>- Computing the fitness</td>
<td></td>
</tr>
<tr>
<td>genemain</td>
<td>- Standard genetic algorithm function</td>
<td></td>
</tr>
<tr>
<td>wsprgene</td>
<td>- Using GA in WSPR synthesis</td>
<td></td>
</tr>
<tr>
<td>Geometry</td>
<td>Algorithm</td>
<td></td>
</tr>
<tr>
<td>spr_redcut</td>
<td>- Main function</td>
<td></td>
</tr>
<tr>
<td>wspr_inter</td>
<td>- compute the intersection of WSPR</td>
<td></td>
</tr>
<tr>
<td>- regions using order_reduced</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- in the ellipse domain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMIs</td>
<td>Algorithm</td>
<td></td>
</tr>
<tr>
<td>wsprlmi</td>
<td>- Using LMIs in SPR synthesis</td>
<td></td>
</tr>
<tr>
<td>wsprlmi</td>
<td>- Using LMIs in WSPR synthesis</td>
<td></td>
</tr>
<tr>
<td>Complete Polynomials Discrimination System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>contest</td>
<td>Determine whether the polynomials</td>
<td></td>
</tr>
<tr>
<td>- set a and polynomial b form SPR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- or WSPR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>compoly</td>
<td>- Using the Complete Discrimination</td>
<td></td>
</tr>
<tr>
<td>- System for Polynomials to determine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- whether the polynomial has real root</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miscellaneous functions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>monic</td>
<td>- Monic the polynomial set matrix</td>
<td></td>
</tr>
<tr>
<td>hurwitz</td>
<td>- Verifying the H-Stability</td>
<td></td>
</tr>
<tr>
<td>sabsole</td>
<td>- Get the square of a polynomial</td>
<td></td>
</tr>
<tr>
<td>- in jω plane</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wspr2spr</td>
<td>- Find SPR solution using WSPR</td>
<td></td>
</tr>
</tbody>
</table>

(1) Function spr_redcut
This function solves the SPR synthesis problem using the geometric algorithm with order reduction.
Syntax: 
\[ [r, b] = \text{spr_redcut}(a, \text{step}) \]
Input:
- \( a \): the matrix stands for the input vertices of polynomials, and each row of this matrix represents a polynomial.
- \( \text{step} \): 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: 
\[ [r, b] = \text{sprgene}(a, \text{ValMulti}, \text{options}) \]
Input:
- \( a \): the matrix stands for the input vertices of polynomials, and each row of this matrix represents a polynomial.
- \( \text{ValMulti} \): the multiple of the upper of coefficients of the search space which is used in WSPR synthesis.
- \( \text{options} \): the vector holds some basic arguments needed by standard GA.

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

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

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 \( \mathcal{F} \) is robustly Hurwitz stable. To solve this SPR synthesis problem with the SPRSt, the solver function spr_redcut 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 \( \mathcal{F} \) 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 \( \mathcal{F} \) is robustly Hurwitz stable. Invoking the solver function sprlmi yields the following result:
the output: 
find a spr polynomial b(s):
1 9.6947 234.6732 150.4671 4.4695

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 $\mathcal{F}$ is robustly Hurwitz stable. Invoking the solver function wsprlmi yields the following result:

The output:

```plaintext
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