POLYNOMIAL METHODS AND LMI OPTIMIZATION: NEW ROBUST CONTROL FUNCTIONS FOR THE POLYNOMIAL TOOLBOX 3.0

Didier Henrion[1,2,4] **Michael Šebek**[3]

[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
[2]Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 182 08 Praha, Czech Republic
[3]Center for Applied Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague
Technická 2, 166 27 Praha 6, Czech Republic
[4]Corresponding author. FAX: +33 5 61 33 69 69. E-mail: `henrion@laas.fr`

Keywords: Linear Systems, Robust Control, Polynomial Methods, Linear Matrix Inequalities, Computer-Aided Control System Design

Abstract

Based on recent robust control results mixing polynomial techniques and LMI optimization, a bunch of new functions implemented in version 3.0 of the Polynomial Toolbox for Matlab are described with the help of illustrative numerical examples.

1 Introduction

Several powerful Matlab-based computer-aided control system design packages are now available on the market. Supported by European projects and networks of excellence [21, 8], the Fortran SLICOT library [21] incorporates mostly state-space methods, whereas the Polynomial Toolbox [24] can be viewed as an alternative, or complementary product including various routines to deal with algebraic entities such as polynomials and polynomial matrices, based on the theory pioneered in [18]. Both packages rely on efficient and reliable numerical linear algebra algorithms.

In the field of polynomial methods for control systems, there has been recently a surge of interest in algorithms based on convex optimization, and most precisely optimization over linear matrix inequalities (LMIs), also called semidefinite programming [7, 29]. New theoretical results emerged from mixing polynomial techniques and LMI optimization, providing new insights especially in the area of robust control.

The objective of this paper is to present a set of new functions incorporated into the new release 3.0 of the Polynomial Toolbox for Matlab [24], based on these recent theoretical achievements. In this paper, the focus in not on the theory, but on the practical use and capabilities of the functions. The reader interested in the theory behind the functions is referred to the technical literature.

The new functions use optimization over polynomials and LMIs to solve various robust control problems, namely:

- robustness analysis:
  - `ptopana` - robust stability analysis of a polytope of polynomial matrices
  - `elliana` - robust stability radius of an ellipsoid of continuous-time scalar polynomials
  - `ellista` - ellipsoidal approximation of the stability domain in the coefficient space of a polynomial
- robust design:
  - `ptopdes` - robust stabilization of a polytope of scalar polynomials
  - `ellides` - robust stabilization of an ellipsoid of scalar polynomials
  - `ptopdes2` - robust proportional-derivative stabilization of a polytope of second-order systems
  - `ellides2` - robust proportional-derivative stabilization of an ellipsoid of second-order systems
  - `sofss` - simultaneous stabilization by scalar static output feedback

LMI problems are solved with the semidefinite programming feature of solver SeDuMi [26]. LMI problems are transformed into semidefinite programs with a user-friendly interface to SeDuMi [23].

See [16] for the full version of this document, including comprehensive function syntax descriptions, and more illustrative numerical examples.

2 Robustness analysis

We consider linear systems represented by fractions of polynomials or polynomial matrices, and affected by parametric (or structured) uncertainty [1]. The objective is to determine if the uncertain system, or equivalently the parametric polynomial matrix, remains stable for all possible values of the uncertainty.

2.1 Function `ptopana`

Function `ptopana` checks robust stability of a polytope of polynomial matrices. Given a set of polynomial matrix vertices

$A_i(s)$ for $i = 1, 2, \ldots$ the function attempts to prove stability of the uncertain polynomial matrix

$$A(s) = \sum_i \lambda_i A_i(s), \quad \sum_i \lambda_i = 1, \quad \lambda_i \geq 0.$$

The underlying theory can be found in [10] and [11].

Mechanical system

Consider the mechanical system represented in Figure 1, whose differential equations after application of the Laplace transform are given by

$$\begin{bmatrix} \begin{pmatrix} m_1 s^2 + d_1 s \\ +c_1 + c_{12} \end{pmatrix} & -c_{12} \\ -c_{12} & \begin{pmatrix} m_2 s^2 + d_2 s \\ +c_2 + c_{12} \end{pmatrix} \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} = \begin{bmatrix} 0 \\ u(s) \end{bmatrix}.$$

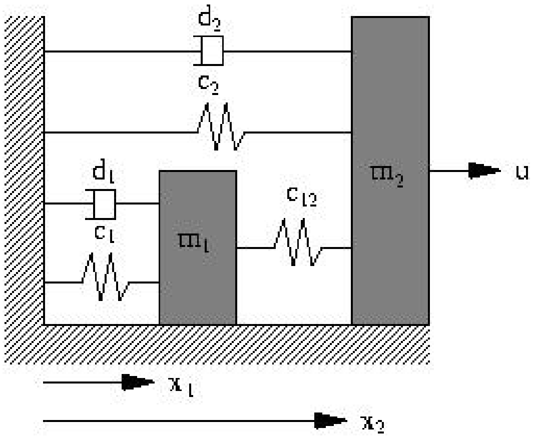

Figure 1: Mechanical system.

We assume that system parameters m_1, d_1, c_1, m_2, d_2, c_2 belong to the uncertainty hyper-rectangle $[1, 3] \times [0.5, 2] \times [1, 2] \times [2,\ 5] \times [0.5,\ 2] \times [2,\ 4]$ and we set $c_{12} = 1$. This mechanical system is passive so it must be open-loop stable (when $u(s) = 0$) independently of the values of the masses, springs, and dampers. However, it is a non-trivial task to know whether the open-loop system is robustly stable in some stability region

$$\mathcal{D} = \{s \in \mathbb{C} : \begin{bmatrix} 1 \\ s \end{bmatrix}^\star S \begin{bmatrix} 1 \\ s \end{bmatrix} < 0\}$$

ensuring a certain damping, where 2×2 matrix S must be speficied. Here we choose the disk of radius 12 centered at -12:

$$\mathcal{D} = \{s \in \mathbb{C} : (s + 12)^2 < 12^2\},$$

i.e. we set

$$S = \begin{bmatrix} 0 & 12 \\ 12 & 1 \end{bmatrix}$$

as the stability matrix. The robust stability analysis problem amounts then to assessing whether the second degree polynomial matrix is robustly stable in $\mathcal{D}$ for all admissible uncertainty. This is an interval polynomial matrix with $m = 2^6 = 64$ vertices. Function `ptopana` proves robust stability as shown in the following script:

```
>> c12 = 1;
>> A = cell(1,2^6); i = 1;
>> for m1 = [1 3], for d1 = [0.5 2],
   for c1 = [1 2], for m2 = [2 5],
   for d2 = [0.5 2], for c2 = [2 4],
    A0 = [c1+c12 -c12; -c12 c2+c12];
    A1 = [d1 0;0 d2]; A2 = [m1 0;0 m2];
    A{i} = pol([A0 A1 A2],2); i = i+1;
   end; end; end;
   end; end; end;
>> S = [0 12; 12 1];
>> ptopana(A,S)
ans =
     1
```

Therefore, the root-locus of the polynomial matrix remains in disk $\mathcal{D}$ for all admissible uncertainty. In Figure 2 we represent the roots of the 64 polynomial matrix vertices.

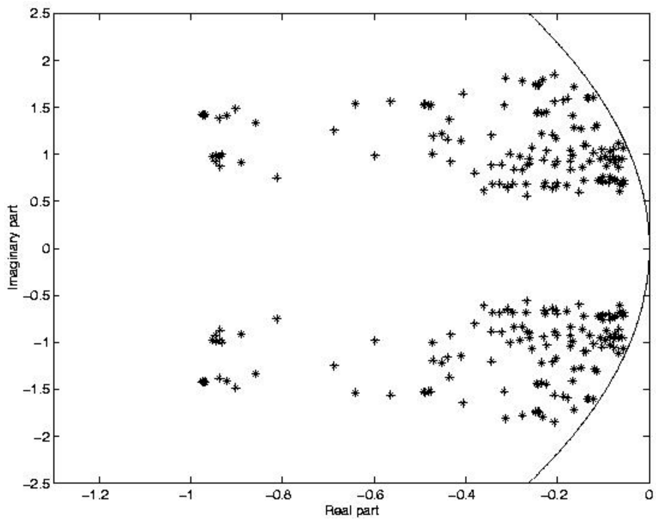

Figure 2: Roots of the 64 polynomial matrix vertices of the mechanical system.

2.2 Function `elliana`

Function `elliana` computes the largest radius r such that the continuous-time ellipsoid of polynomials

$$\{p(s, q) = p_0(s) + \sum_{i=1}^{n} q_i p_i(s),\ q'q \leq r^2\},$$

remains robustly stable. In the above description, $p_0(s)$ is a given stable nominal polynomial, $p_i(s)$ are given polynomials of degree less than or equal to the degree of $p_0(s)$, and q is a real vector with entries q_i modelling the uncertainty, see [1, §7.2]. Note that even though LMIs can be invoked to solve numerically the problem, it is the analytic formulation of the solution found in [1, §7.2] that has been implemented in function `elliana`.

Example

Consider as in [1, Example 7.2] the ellipsoid of polynomials described by

$$\begin{array}{rcl} p_0(s) & = & 129+166s+237s^2+108s^3+80s^4 \\ p_1(s) & = & -16+24s-12s^2+4s^3 \\ p_2(s) & = & -21+42s-21s^2 \end{array}$$

We obtain a stability radius of $r = 1.1125$ with the following script:

```
>> r = elliana(129+166*s+237*s^2+108*s^3+80*s^4,...
       [-16+24*s-12*s^2+4*s^3, -21+42*s-21*s^2])
r =
    1.1125
```

2.3 Function `ellista`

Function `ellista` builds an inner ellipsoidal approximation of the (generally non-convex) stability domain in the space of coefficients of a monic polynomial, based on the theoretical results of [12]. This ellipsoidal approximation can be used for robust design as well.

Third-degree discrete-time stability domain

In the three-dimensional space of coefficients of third degree discrete-time monic polynomials, the exact stability domain is a non-convex set delimited by two triangles $V_1V_2V_3$, $V_2V_3V_4$ of vertices

$$p_{V_1} = \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix} p_{V_2} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} p_{V_3} = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} p_{V_4} = \begin{bmatrix} -1 \\ 3 \\ -3 \end{bmatrix}$$

supporting a hyperbolic paraboloid with saddle point

$$p_S = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Function `ellista` computes an ellipsoidal inner approximation of the stability domain containing a stable polynomial with all roots at the origin, as shown in Figure 3, where the actual non-convex stability domain is also represented.

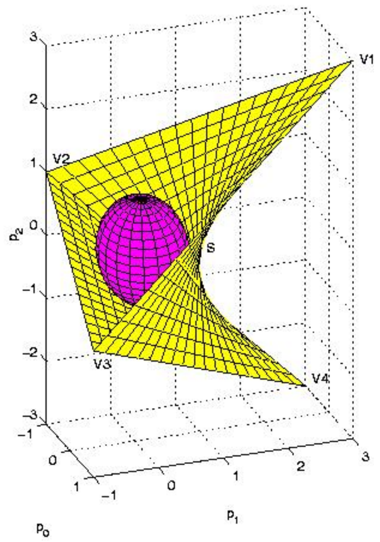

Figure 3: Hyperbolic and ellipsoidal stability domains for a third-degree discrete-time polynomial.

3 Robust design

Except function `sofss`, all the robust design functions described below are based on an LMI inner approximation of the stability domain in the space of polynomial matrix coefficients. The LMI approximation is obtained from results on strictly positive real rational functions and positive polynomial matrices. The LMI stability domain is built around a reference, or central polynomial matrix. In order to perform design, the functions therefore require a central polynomial matrix as an input argument.

3.1 Function `ptopdes`

This function attempts to stabilize a polytope of scalar plants with a fixed-order compensator. We consider a proper scalar plant

$$\frac{b(s,q)}{a(s,q)}$$

whose denominator and numerator polynomials are affected by polytopic uncertainty. The components of uncertainty parameter vector q belong to a polytope with given vertices q^i, i.e.

$$q = \sum_i \lambda_i q^i, \quad \sum_i \lambda_i = 1, \quad \lambda_i \geq 0.$$

We are seeking a controller

$$\frac{y(s)}{x(s)}$$

of fixed order with monic denominator polynomial. The controller is settled in a standard negative feedback configuration. Equivalently, polynomials $x(s)$ and $y(s)$ are sought such that the roots of polytopic characteristic polynomial

$$d(s,q) = a(s,q)x(s) + b(s,q)y(s)$$

remain in the stability region for all admissible values of uncertain parameter q in the polytope.

Robot

We consider the problem of designing a robust controller for the approximate ARMAX model of a PUMA 762 robotic disk grinding process [28]. From the results of identification and because of the nonlinearity of the robot, the coefficients of the numerator of the plant transfer function change for different positions of the robot arm. We consider variations of up to 20%

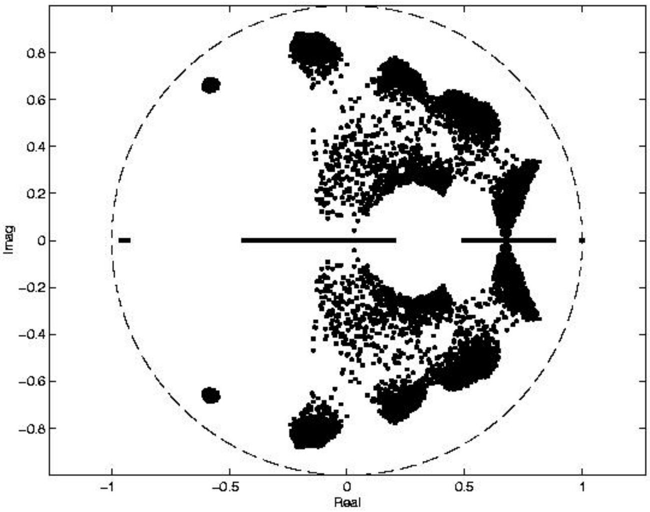

Figure 4: Robot. Robust root locus.

around the nominal value of the parameters. The fourth-order discrete-time model is given by

$$\frac{b(z^{-1},q)}{a(z^{-1},q)} = \frac{\begin{pmatrix} (0.0257+q_1)+(-0.0764+q_2)z^{-1} \\ +(-0.1619+q_3)z^{-2}+(-0.1688+q_4)z^{-3} \end{pmatrix}}{\begin{pmatrix} 1-1.914z^{-1}+1.779z^{-2} \\ -1.0265z^{-3}+0.2508z^{-4} \end{pmatrix}}$$

where

$$|q_1| \le 0.00514,\ |q_2| \le 0.01528,$$
$$|q_3| \le 0.03238,\ |q_4| \le 0.03376.$$

The characteristic polynomial of the closed-loop system is given by

$$d(z,q) = z^{12}[(1-z^{-1})a(z^{-1},q)x(z^{-1})+z^{-5}b(z^{-1},q)y(z^{-1})]$$

where the term $1-z^{-1}$ is introduced in the controller denominator to maintain the steady state error to zero when parameters are changed. In order to use function `ptopdes` we must provide a reference closed-loop polynomial, or central polynomial around which robust design will be carried out, see [15] for more details. With the input central polynomial

$$c(z) = z^{19}$$

function `ptopdes` finds the seventh-order robust controller

$$\frac{y(z^{-1})}{x(z^{-1})} = \frac{\begin{pmatrix} -0.2863+0.2928z^{-1}+0.0221z^{-2} \\ -0.1558z^{-3}+0.0809z^{-4}+0.1420z^{-5} \\ -0.1254z^{-6}+0.0281z^{-7} \end{pmatrix}}{\begin{pmatrix} 1+1.1590z^{-1}+0.9428z^{-2} \\ +0.4996z^{-3}+0.3044z^{-4}+0.4881z^{-5} \\ +0.4003z^{-6}+0.3660z^{-7} \end{pmatrix}}.$$

The robust root locus, obtained by taking 1000 random plants within the uncertainty polytope, is represented in Figure 4.

3.2 Function `ellides`

This function attempts to stabilize a scalar plant affected by ellipsoidal uncertainty with a fixed-order compensator. We consider a proper scalar plant

$$\frac{b(s,q)}{a(s,q)} = \frac{b^0(s)+q'b^1(s)}{a^0(s)+q'a^1(s)}$$

whose denominator and numerator polynomials are affected by norm-bounded uncertainty: real parameter vector q satisfies $q'q \le 1$ and $a^1(s)$, $b^1(s)$ are polynomial row vectors of the same dimension as q.

We are seeking a controller

$$\frac{y(s)}{x(s)}$$

of fixed order with monic denominator polynomial. The controller is settled in a standard negative feedback configuration. Equivalently, polynomials $x(s)$ and $y(s)$ are sought such that the roots of polytopic characteristic polynomial

$$d(s,q) = a(s,q)x(s)+b(s,q)y(s)$$

remain in the stability region for all q such that $q'q \le 1$.

Mixing tanks

We consider the two mixing tanks arranged in cascade with recycle stream shown in Figure 5 and described in [6]. The

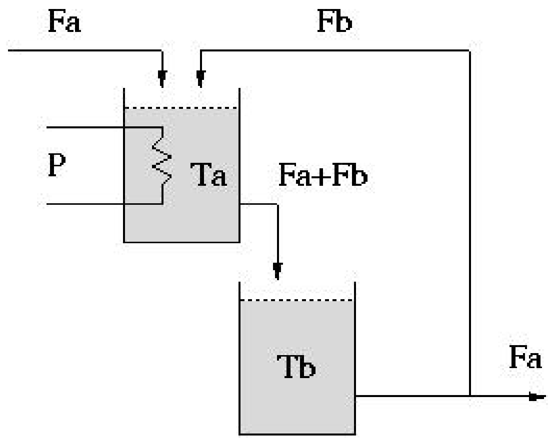

Figure 5: Two-tank system.

controller must be designed to maintain the temperature T_b of the second tank at a desired set point by manipulating the power P delivered by the heater located in the first tank. The only available measurement is temperature T_b. The identification of the nominal plant model is carried out using a standard least-squares method [6]. The discrete-time nominal plant is given by

$$\frac{b^0(z)}{a^0(z)} = \frac{b_0^0+b_1^0 z}{a_0^0+a_1^0 z+z^2}$$

with nominal plant vector

$$p_c = \begin{bmatrix} b_0^0 \\ b_1^0 \\ a_0^0 \\ a_1^0 \end{bmatrix} = \begin{bmatrix} 0.0038 \\ 0.0028 \\ 0.2087 \\ -1.1871 \end{bmatrix}.$$

An ellipsoidal uncertainty model is readily available as a by-product of the least-squares identification technique [6]. Uncertainty polynomial vectors are given by

$$a^1(s) = 10^{-2} \begin{bmatrix} -0.05575 + 0.03987z \\ 0.0002497 - 0.02517z \\ 19.73 - 13.77z \\ -13.77 + 19.62z \end{bmatrix}$$

and

$$b^1(s) = 10^{-3} \begin{bmatrix} 2.036 - 0.02397z \\ -0.02397 + 2.037z \\ -0.5575 + 0.002497z \\ 0.3987 - 0.2517z \end{bmatrix}.$$

Now suppose that we are seeking a first-order controller

$$\frac{y_0 + y_1 z}{x_0 + z}$$

robustly stabilizing the plant for all admissible models within the uncertainty ellipsoid.

Similarly to function `ptopdes`, function `ellides` requires a reference closed-loop polynomial, or central polynomial around which robust design could be carried out. With the choice

$$c(z) = (0.1 + z)^3$$

as an input central polynomial, function `ellides` returns

$$\frac{y(z)}{x(z)} = \frac{6.068 + 6.981z}{0.3524 + z}$$

as a first-order robustly stabilizing controller.

The robust root-locus of closed-loop characteristic polynomial obtained by describing randomly the uncertainty ellipsoid is represented in Figure 6. We can check that indeed all characteristic polynomial roots stay in the unit disk for all admissible uncertainty.

Enforcing now pole location in the stability region

$$\mathcal{D} = \{z \in \mathbb{C} : |z| \leq 0.7\},$$

i.e. setting

$$S = \begin{bmatrix} -(0.7)^2 & 0 \\ 0 & 1 \end{bmatrix}$$

as the stability matrix in function `ellides`, we obtain

$$\frac{y(z)}{x(z)} = \frac{-35.69 + 137.3z}{0.5913 + z}$$

and the robust root-locus of Figure 7.

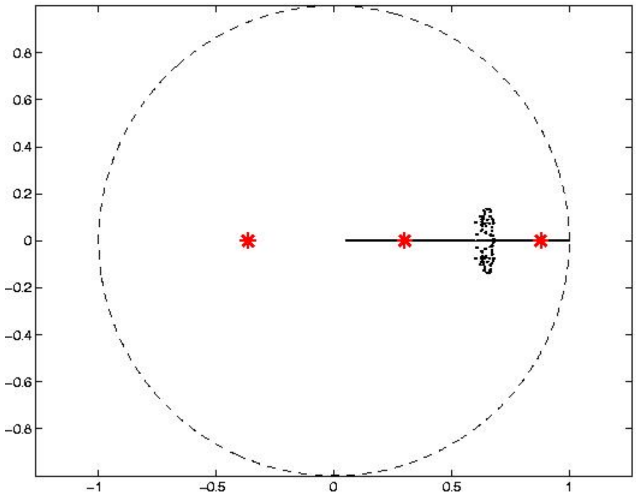

Figure 6: Root-locus within the unit disk.

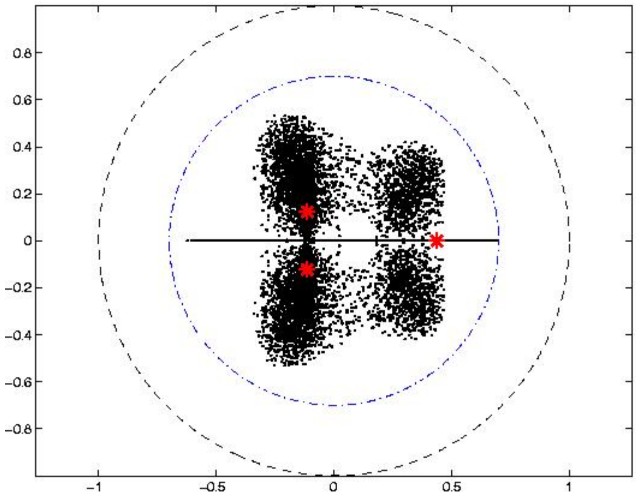

Figure 7: Root-locus within $|z| \leq 0.7$.

3.3 Function `ptopdes2`

Consider the multivariable linear system described by the second-order dynamical equations

$$\begin{aligned} (A_0 + A_1 s + A_2 s^2)x &= Bu \\ y &= Cx \end{aligned}$$

controlled by a proportional-derivative (PD) output-feedback controller of the form

$$u = -(F_0 + F_1 s)y$$

so that the closed-loop system behavior is captured by the quadratic polynomial matrix

$$N(s) = (A_0 + BF_0C) + (A_1 + BF_1C)s + A_2 s^2.$$

We assume that the second-order system is affected by polytopic uncertainty, i.e. quadratic matrix $A(s) = A_0 + A_1 s + A_2 s^2$ belongs to a polytope with given polynomial matrix vertices $A^1(s),\ A^2(s)\ , \ldots$

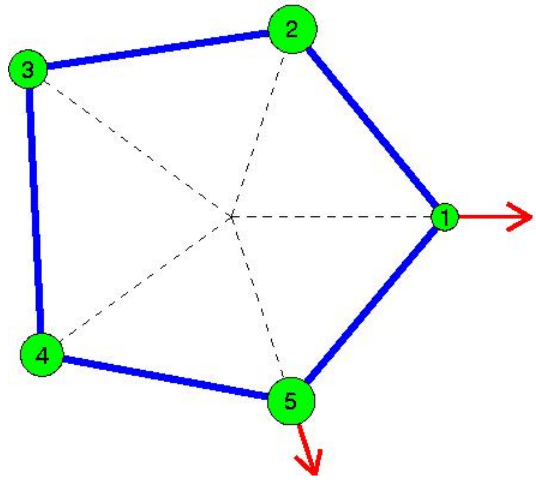

Figure 8: Five masses linked by elastic springs.

Point	Mass	Distance	Spring	Stiffness
1	0.5093	0.8034	1-2	1.461
2	0.9107	0.7430	2-3	1.369
3	0.7224	0.9456	3-4	1.088
4	0.8077	0.8810	4-5	1.203
5	0.8960	0.7282	5-1	1.468

Table 1: System data.

Mechanical structure

We consider the mechanical system shown in Figure 8, consisting of five material points linked by elastic springs [3]. The points can slide without friction along their respective axes. Mass, distance to the origin at the equilibrium, and spring stiffness are given for each point in Table 1.

The system is controlled by two external forces acting at masses 1 and 5. System matrices are given in the script below. Open-loop poles are all purely imaginary and located at $\pm i1.783$, $\pm i1.380$, $\pm i1.145$, $\pm i0.5675$ and $\pm i0.3507$.

A stabilizing PD controller is obtained in [3] with a nearly optimal linear-quadratic robust design method, yielding matrices F_0^0 and F_1^0 given in the script below. Poles of closed-loop quadratic matrix polynomial

$$D(s) = (A_0 + BF_0^0 C) + (A_1 + BF_1^0 C)s + A_2 s^2$$

are located at $-0.1067 \pm i1.406$, -0.1405, $-0.1809 \pm i0.5350$, $-0.2174 \pm i1.099$, $-0.8157 \pm i1.450$ and -1.016. Feedback matrix $F^0 = [F_0^0 \quad F_1^0]$ has norm $f^0 = 1.859$.

In view of the closed-loop poles, we choose

$$\mathcal{D} = \{s \in \mathbb{C} : \mathrm{Re}\, s < -0.1\}$$

as the stability region. With the above polynomial matrix $D(s)$ as central system matrix, we invoke function `ptopdes2` as follows:

```
>> A2 = eye(5); A1 = zeros(5);
>> A0 = [2.5647 1.0797 0 0 1.0890
         0.6038 0.8206 0.4766 0 0
         0 0.6009 1.5044 0.4808 0
         0 0 0.4300 1.1142 0.5131
         0.6190 0 0 0.4626 0.8352];
>> A = pol([A0 A1 A2],2);
>> B = [0 1.9637;0 0;0 0;0 0;1.1161 0];
>> F00 = -[-0.0396 0.0220 -0.3685 -0.8069 -0.4099
          -0.3993 -0.6453 -0.4886 -0.2269 -0.0322];
>> F01 = [0.0152 -0.3694 0.0647 -0.0498 1.3167
         1.1859 -0.5896 -0.2165 -0.3263 0.0268];
>> D = pol([A0+B*F00 A1+B*F01 A2],2);
>> S = [0.2 1;1 0];
>> [F0,F1] = ptopdes2(A,B,[],D,S)
```

Running the above script, we obtain feedback matrices:

$$F_0 = \begin{bmatrix} -0.1610 & -0.1136 & -0.03508 & 0.08337 & -0.05075 \\ -0.2706 & 0.04941 & 0.1440 & 0.06144 & -0.1366 \end{bmatrix},$$
$$F_1 = \begin{bmatrix} -0.1169 & -0.3153 & 0.2319 & 0.1873 & 0.5418 \\ 0.5383 & -0.2237 & -0.004471 & -0.2137 & 0.06340 \end{bmatrix}.$$

Poles of the new closed-loop quadratic matrix polynomial

$$N(s) = (A_0 + BF_0 C) + (A_1 + BF_1 C)s + A_2 s^2$$

are located at $-0.1090 \pm i1.404$, $-0.1348 \pm i0.5436$, $-0.1445 \pm i1.1110$, $-0.1823 \pm i0.2301$ and $-0.2603 \pm i1.457$, well inside region $\mathcal{D}$. Feedback matrix $F = [F_0 \quad F_1]$ has largest singular value $f = 0.7593 < f^0$. Consequently, new feedback F requires less control effort and is less prone to saturation than original feedback F^0.

3.4 Function `ellides2`

Now we assume that the second-order system introduced in the description of function `ptopdes2` is affected by ellipsoidal uncertainty. In other words, quadratic matrix $A(s)$ is subject to additive norm-bounded (unstructured) uncertainty

$$A(s) = A^0(s) + \Delta A^1(s), \quad \sigma_{\max}(\Delta'\Delta) \leq 1$$

where Δ is a uncertainty matrix of arbitrary column dimension, $A^0(s)$ is the nominal quadratic system matrix, $A^1(s)$ is the quadratic uncertainty matrix, and $\sigma_{\max}$ denotes the maximum singular value.

Wing in airstream

In [27] the authors consider an eigenvalue problem arising from the analysis of the oscillations of a wing in an airstream. Quadratic system matrix coefficients are given in the script below. The system is open-loop unstable since its poles are located at $0.09427 \pm i2.553$, $-0.8848 \pm i8.442$ and $-0.9180 \pm i1.761$.

We choose $\mathcal{D} = \mathcal{D}^1 \cap \mathcal{D}^2$ with

$$\mathcal{D}^1 = \{s \in \mathbb{C} : -\mathrm{Re}\, s < 0\}, \quad \mathcal{D}^1 = \{s \in \mathbb{C} : -2 < -\mathrm{Re}\, s\}$$

as the stability region, i.e. we set

$$S^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad S^2 = \begin{bmatrix} -4 & -1 \\ -1 & 0 \end{bmatrix}$$

as stability matices. Let $D(s) = (s+1)^2 I_3$ be the central closed-loop matrix with roots in $\mathcal{D}$. Macro `ellides2` called with no uncertainty matrix returns the following stabilizing feedback matrices:

```
>> A2 = [17.6 1.28 2.89; 1.28 0.824 0.413;
         2.89 0.413 0.725];
>> A1 = [7.66 2.45 2.1; 0.23 1.04 0.223;
         0.6 0.756 0.658];
>> A0 = [121 18.9 15.9; 0 2.7 0.145;
         11.9 3.64 15.5];
>> A = pol([A0 A1 A2],2);
>> D = diag([(s+1)^2 (s+1)^2 (s+1)^2]);
>> S = cell(2,1);
>> S{1} = [0 1;1 0]; % Re(s) < 0
>> S{2} = [-4 -1;-1 0]; % Re(s) > -2
>> [F0,F1] = ellides2(A,[],[],[],D,S)
F0 =
   -4.8884  -13.1087   -2.4724
    1.9863   -1.0819    0.9174
    1.5609   -1.3376  -12.9224
F1 =
   14.0911   -3.9281    0.0044
    1.5266    0.6136    0.6725
    1.0762   -0.7899   -0.2328
>> N = pol([A0+F0 A1+F1 A2],2);
>> roots(N)
ans =
  -1.0498 + 2.6799i
  -1.0498 - 2.6799i
  -0.7443 + 1.5821i
  -0.7443 - 1.5821i
  -0.7736
  -0.4570
```

If a failure affects the second actuator, function `ellides2` is still able to compute a stabilizing PD feedback:

```
>> B = [1 0;0 0;0 1]; C = eye(3);
>> [F0,F1] = ellides2(A,[],B,C,D,S)
F0 =
   -4.3948  -15.1142   -0.4306
    1.9550   -1.5836  -12.8369
F1 =
   14.9815   -3.9831    0.7908
    2.1173   -0.4122    0.4698
>> N = pol([A0+B*F0*C A1+B*F1*C A2],2);
>> roots(N)
ans =
  -1.0335 + 2.6151i
  -1.0335 - 2.6151i
  -0.3391 + 1.7762i
  -0.3391 - 1.7762i
  -1.8296 + 0.5039i
  -1.8296 - 0.5039i
```

Similarly, assuming that all the actuators are available, but that a failure affects the second sensor, function `ellides2` returns

```
>> B = eye(3); C = [1 0 0;0 0 1];
>> [F0,F1] = ellides2(A,[],B,C,D,S)
F0 =
   -4.7800    3.0768
   13.5196    2.4315
    5.2709  -11.7931
F1 =
   15.7655    1.5617
    4.5934    0.7399
    3.2495    0.7613
>> N = pol([A0+B*F0*C A1+B*F1*C A2],2);
>> roots(N)
ans =
  -0.9126 + 2.6002i
  -0.9126 - 2.6002i
  -1.9667
  -1.3883
  -0.3567 + 0.4326i
  -0.3567 - 0.4326i
```

Finally, we suppose that the damping matrix is subject to additive norm-bounded uncertainty, i.e. $A^1(s) = \delta s I_3$. Function `ellides2` was then able to robustly stabilize the system:

```
>> A1 = 0.19*s*eye(3);
>> [F0,F1] = ellides2(A,A1,[],[],D,S)
F0 =
   -4.7768  -15.0286   -0.0750
    1.1389   -1.0147    0.3262
    2.9717   -2.8173  -13.0886
F1 =

   14.5147   -3.2621    0.2163
    0.3704    0.7278    0.3836
    2.1353   -0.3827    0.0369
```

for all uncertainty with worst-case norm $\delta = 0.19$. In Figure 9 we represent the closed-loop robust root locus for 10000 randomly chosen systems in the admissible uncertainty range. Nominal closed-loop poles are represented by red stars.

3.5 Function `sofss`

Let

$$\frac{b_i(s)}{a_i(s)}, \quad i = 1, 2 \ldots$$

denote a set of scalar plants, where $a_i(s)$ and $b_i(s)$ are scalar polynomials of degree n. Function `sofss` solves the problem of finding a scalar static feedback gain k that simultaneous stabilizes the plants, i.e. such that the roots of all the characteristic polynomials

$$c_i(s) = a_i(s) + k b_i(s), \quad i = 1, 2 \ldots$$

belong to the left half plane. Even though the problem can be solved using LMIs, function `sofss` solves the problem with standard numerical algebra, based on the results of [14].

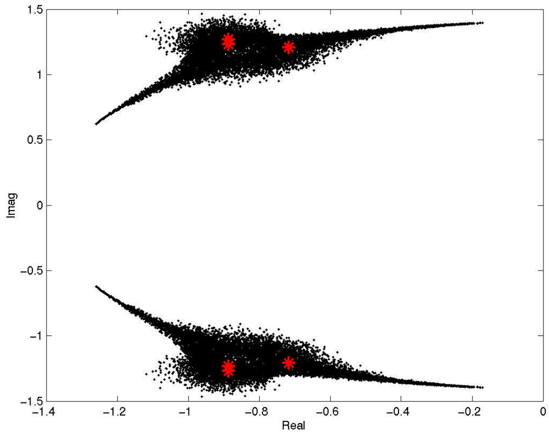

Figure 9: Robust root-locus of the wing.

Reactor

Consider the continuous stirred tank reactor model studied in [17]. The non-linear model is

$$\begin{aligned} x_1 &= (x_2+0.5)\exp(Ex_1/(x_1+2)) - (2+u)(x_1+0.25) \\ x_2 &= 0.5 - x_2 - (x_2+0.5)\exp(Ex_1/(x_1+2)) \end{aligned}$$

where E is a parameter related to the activation energy. During the life of the reactor, some representative values of E are 20, 25 and 30. Assuming that only $y = x_1$ is available for feedback, the $N = 3$ linearized systems of order $n = 2$ to be simultaneously stabilized are given by

$$\begin{aligned} b_1(s)/a_1(s) &= (0.5-0.25s)/(11-5s+s^2) \\ b_2(s)/a_2(s) &= (-0.5-0.25s)/(-2.25-2.25s+s^2) \\ b_3(s)/a_3(s) &= (-0.5-0.25s)/(-3.5-3.5s+s^2). \end{aligned}$$

Calling function `sofss` with the following script

```
>> b1=0.5-0.25*s;a1=11-5*s+s^2;
>> b2=-0.5-0.25*s;a2=-2.25-2.25*s+s^2;
>> b3=-0.5-0.25*s;a3=-3.5-3.5*s+s^2;
>> sofss({a1 a2 a3},{b1 b2 b3})
ans =
  -22.0000  -20.0000
```

we obtain that the three plants are simultaneously stabilizable by a static output feedback $u = ky$ for any value of k such that $-22 < k < -20$.

4 Conclusion

In this paper we have presented a set of new Matlab functions from release 3.0 of the Polynomial Toolbox [24]. We are planning to extend the function in various directions – see our list in [16, §12] – depending mainly on feedback and requests by users.

There is still a large amount of new results on LMI optimization over polynomials and polynomial matrices lacking from practical implementations. Recently, theoretical results on positive polynomials and sum-of-squares decompositions [20, 19, 22] opened up new avenues for the development of computer-aided control system design packages. In the area of control, potential applications include stability analysis and design for uncertain and/or non-linear systems using polynomial Lyapunov functions [4, 5] or LMI relaxations for difficult non-convex optimization problems arising in robustness analysis and design [22]. Preliminary work in this direction already resulted in two complementary packages called GloptiPoly [13] and SOSTOOLS [25], for which several extensions are currently being developped. Tailored, computationally efficient and numerically reliable algorithms for convex optimization over polynomials and polynomial matrices are also being investigated [20, 9, 2].

Acknowledgments

This work was supported by the Grant Agency of the Czech Republic under Project No. 102/02/0709, by the Czech Ministry of Education Project No. LN00B096, and by the CNRS STIC Departement Young Researcher Grant No. 01N80/0474. Comments and help by Zdeněk Hurák, Michal Kvasnica and Thomas Gauchet were appreciated.

References

[1] J. Ackermann. Robust Control. Systems with Uncertain Physical Parameters. *Springer Verlag*, Berlin, 1993.

[2] B. Alkire, L. Vandenberghe. Convex optimization problems involving finite autocorrelation sequences. *Research Report*, University of California at Los Angeles, 2002. See `www.ee.ucla.edu/~vandenbe`

[3] F. D. Barb, A. Ben-Tal, A. Nemirovski. Robust dissipativity of interval uncertain systems. *Research Report*, Delft University of Technology, The Netherlands, 2001.

[4] P. A. Bliman. A Convex Approach to Robust Stability for Linear Systems with Uncertain Scalar Parameters. *Research Report*, No. 4316, INRIA Rocquencourt, France, 2001. See `www.inria.fr/rrrt/rr-4316`

[5] G. Chesi, A. Garulli, A. Tesi, A. Vicino. Homogeneous Lyapunov functions for systems with structured uncertainties. *Research Report*, Università degli Studi di Siena, Italy, 2002. See `www.dii.unisi.it/~garulli`

[6] O. D. Crisalle, H. M. Mahon, D. Bonvin. Study of Robust Control Designs using the Critical Direction Method for Ellipsoidal Uncertainties. *Proceedings of the IEEE Southcon94 Conference*, pp. 173–180, Orlando, Florida, 1994.

[7] L. El Ghaoui, S. I. Niculescu (Editors). Advances in Linear Matrix Inequality Methods in Control. *SIAM*, Philadelphia, 1999.

[8] EUROPOLY: The European Network of Excellence for Industrial Applications of Polynomial Design Methods. Network No. CP97:7010 funded by the European Commission Inco/Copernicus Programme. 1998-2001. See `www.utia.cas.cz/europoly`

[9] Y. Genin, Y. Hachez, Y. Nesterov, P. Van Dooren. Optimization problems over positive pseudo-polynomial matrices. *Research*

Report, Université Catholique de Louvain, Belgium, 2001. See www.auto.ucl.ac.be/∼vdooren

[10] D. Henrion, D. Arzelier, D. Peaucelle, M. Šebek. An LMI Condition for Robust Stability of Polynomial Matrix Polytopes. *IFAC Automatica*, Vol. 37, pp. 461–468, 2001.

[11] D. Henrion, O. Bachelier, M. Šebek. D-Stability of Polynomial Matrices. *International Journal of Control*, Vol. 74, No. 8, pp. 845–856, 2001.

[12] D. Henrion, D. Peaucelle, D. Arzelier, M. Šebek. Ellipsoidal Approximation of the Stability Domain of a Polynomial. *Proceedings of the European Control Conference*, Porto, Portugal, pp.384–389, 2001.

[13] D. Henrion, J. B. Lasserre. GloptiPoly: Global Optimization over Polynomials with Matlab and SeDuMi. *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, 2002. See www.laas.fr/∼henrion/software/gloptipoly

[14] D. Henrion, M. Šebek, V. Kučera. An Algorithm for Static Output Feedback Simultaneous Stabilization of Scalar Plants. *Proceedings of the IFAC World Congress on Automatic Control* Barcelona, Spain, 2002.

[15] D. Henrion, M. Šebek, V. Kučera. Positive Polynomials and Robust Stabilization with Fixed-Order Controllers. *Research Report*, No. 02325, LAAS-CNRS, Toulouse, France, 2002. See www.laas.fr/∼henrion/papers/ptopdes.pdf

[16] D. Henrion, M. Šebek. New Robust Control Functions for the Polynomial Toolbox 3.0. *Research Report*, LAAS-CNRS, Toulouse, France, 2002. See www.laas.fr/∼henrion/papers/polyrobust.pdf

[17] G. D. Howitt, R. Luus. Simultaneous Stabilization of Linear Single-Input Systems by Linear State Feedback Control. *International Journal of Control*, Vol. 54, No. 4, pp. 1015–1030, 1991.

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

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

[20] Y. Nesterov. Squared functional systems and optimization problems. Chapter 17, pp. 405–440 in H. Frenk, K. Roos, T. Terlaky (Editors). High performance optimization. *Kluwer Academic Publishers*, Dordrecht, 2000.

[21] NICONET: The Numerics in Control Network. Project No. BRRT-CT97-5040 funded by the European Commission BRITE-EURAM III Thematic Networks Programme. 1998-2002. See www.win.tue.nl

[22] P. A. Parrilo. Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. *PhD Thesis*, California Institute of Technology, Pasadena, California, 2000. See www.control.ethz.ch/∼parrilo

[23] D. Peaucelle, D. Henrion, Y. Labit. SeDuMi Interface: A user-friendly free Matlab package for defining LMI problems. *Proceedings of the IEEE Conference on Computer-Aided Control System Design*, Glasgow, Scotland, 2002. See www.laas.fr/∼peaucell/SeDuMiInt.html

[24] PolyX, Ltd. The Polynomial Toolbox for Matlab. Prague, Czech Republic. See www.polyx.cz

[25] S. Prajna, A. Papachristodoulou, P. A. Parrilo. SOSTOOLS: Sum of Squares Optimization Toolbox for Matlab. *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, 2002. See www.cds.caltech.edu/sostools

[26] 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 fewcal.kub.nl/sturm/software/sedumi.html

[27] F. Tisseur, N. J. Higham. Structured pseudospectra for polynomial eigenvalue problems, with applications. *SIAM Journal on Matrix Analysis and Applications*, Vol. 23, No. 1, pp. 187–208, 2001.

[28] Y. Tong and N. K. Sinha. A Computational Technique for the Robust Root Locus. *IEEE Transactions on Industrial Electronics*, Vol. 41, No. 1, pp. 79–85, 1994.

[29] H. Wolkowicz, R. Saigal, L. Vandenberghe (Editors). Handbook of Semidefinite Programming. *Kluwer Academic Publishers*, 2000.