

POLMAT: MUPAD LIBRARY FOR POLYNOMIAL MATRICES

Petr Augusta* Zdeněk Hurák**,¹

* *Dept. of Control Engineering, Faculty of Electrical
Engineering, Czech Tech. University in Prague, Czech Rep.
e-mail : augusp1@control.felk.cvut.cz*

** *Center for Applied Cybernetics, Faculty of Electrical
Engineering, Czech Tech. University in Prague, Czech Rep.
e-mail : z.hurak@c-a-k.cz*

Abstract: The paper presents a new freely available library for symbolic computation with polynomial matrices. The library is developed for MuPAD, which is an easily accessible efficient programming language and comfortable environment for symbolic math. The Polmat library is focused on algorithms useful in control and filter design such as solvers for linear equations with polynomials and polynomial matrices and spectral factorisation of polynomial matrices. *Copyright ©2005 IFAC*

Keywords: Computer-aided control system design, matrix polynomial equations, polynomial methods.

1. INTRODUCTION

Polynomials and polynomial matrices turn out convenient tools for description of dynamic systems. Ratios of two univariate polynomials are commonly used for description of linear dynamic systems with single input and single output (SISO). Left or right fractions of polynomial matrices generalise in a straightforward way the concept of transfer functions for systems with multiple inputs and multiple outputs (MIMO). Algebraic manipulations with polynomials and polynomial matrices can reveal various properties of the systems being modelled obviating the numerically intensive simulation, see (Kailath, 1980) for a comprehensive overview. For instance, testing for the location of zeros of a polynomial matrix answers the common question about the stability of MIMO system, testing a polynomial matrix for column reducedness and consequently comparing

column degrees of the two polynomial matrices defining a right polynomial matrix fraction offers itself as a procedure for testing causality of a model, computation of the rank of a polynomial matrix is a step in the controllability and observability analysis. But it is not only the analysis that can be performed by working with polynomial matrices. Solving linear equations with polynomials and polynomial matrices lends itself as a basic computational step in the pole-placement design methodology and when combined with spectral factorisation, it constitutes the basic computational framework for LQG, H_2 and H_∞ -optimal control, see (Kučera, 1979), (Hunt, 1993) and (Grimble and Kučera, 1996) for comprehensive overview.

The polynomial methods for control design now provide:

- (1) an alternative description of linear systems to state-space models,
- (2) methods for analysis of properties of dynamical systems,

¹ Supported by the Ministry of Education of the Czech Republic under Project 1M6840770004.

(3) methods for controller and filter design.

Their advantage is that the input-output character of the model is conserved during the computation and is not lost as in the state-space techniques, which yields the polynomial techniques very natural and intuitive.

However, unlike the state-space techniques, which enjoy the abundance of software libraries (both commercial and free) for manipulating constant matrices like solvers for Sylvester, Lyapunov and Ricatti equations, the polynomial methods rely on highly specialised packages that cannot be found in common computational tools. The only complete library is Polynomial Toolbox for Matlab by (Kwakernaak and Šebek, 1998), which is a commercial toolbox developed and distributed by the PolyX company. Only recently, a few new packages have been developed that slightly extend this offer of tools for polynomial matrices to diverse platforms and languages:

- (1) Polpack++: A C++ library for numerical computation with polynomial matrices by Leoš Halmo (Halmo, 2004),
- (2) *polynomial*: Java package for numerical computation with polynomial matrices by Michal Paděra (Paděra, 2004),
- (3) PTOOL-TI89: C library for numerical computation with polynomial matrices for TI-89/TI-92 programmable calculators by Petr Štefko (Štefko, 2004),
- (4) a few functions for polynomial and rational matrices in Fortran SLICOT library (Benner *et al.*, 1997)
- (5) a few functions for polynomial matrices in Scilab (INRIA and ENPC, 2000)
- (6) a very limited set of (not control related) functions for polynomial matrices in Maple

The authors of the Polmat package² presented in this paper aim to extend this offer with a set of function oriented on symbolic computation with polynomial matrices. The decision was made to build this library atop the MuPAD system (SciFace, 2000), which is a young but very lively commercial project loosely mimicking Maple. A nice feature of MuPAD is its accessibility for individual students and researchers – it is for free for educational and research purposes. The authors believe that this fact might help in spreading the polynomial computation in the control community. The focus of MuPAD on symbolic computation was also appreciated because most of the available tools for polynomial matrices rely on numerical algorithms and therefore suffer from the common problems related to rounding. Symbolic

² Polmat package constitutes one of the outcomes of Petr Augusta Master's degree diploma work at the Dept. of Control Engineering, FEE CTU in Prague (Augusta, 2005)

algorithms, on the other hand, express real numbers as ratios of two integers, and therefore no errors are introduced into the computation. The price is, however, extreme computational burden. Even with this lowered computational efficiency, symbolic computation can be regarded as a helper with the development of numerical algorithms. Not to mention the educational role.

2. INTRODUCTION TO THE POLMAT PACKAGE

This section serves the purpose of introducing the reader to the Polmat package, showing how intuitive and convenient this tool is. The comprehensive manual is available online (Augusta and Hurák, 2005).

2.1 Entering a polynomial matrix

Creating two polynomial matrices

$$A(d) = \begin{pmatrix} d & 1-d \\ 5d^2 & d^2 \end{pmatrix}, \quad B(s) = \begin{pmatrix} 0 & 1-s \\ 1+s & 1-s^2 \end{pmatrix}.$$

in MuPAD command line is as easy as typing

```

>> A:=polmat([[d, -d+1],[5*d^2, d^2]],d)

      +-          +-
      |    d,  - d + 1 |
      |              |
      |    2      2    |
      | 5 d ,    d    |
      +-          +-
>> B:=polmat([[0, 1-s],[1+s, 1-s^2]],s)

      +-          +-
      |    0,  - s + 1 |
      |              |
      |              2 |
      | s + 1, - s + 1 |
      +-          +-

```

2.2 Inquiring about the properties of a polynomial matrix

We perform a test of a matrix row reduction.

```

>> polmat::isreduced(A, d, "row")

      TRUE

```

We compute the leading degree, the leading coefficient and the coefficient of the term s^0 .

```
>> polmat::ldeg(B, s)
                2
>> polmat::lcoeff(B, s)
      +-      +-
      | 0, 0 |
      |      |
      | 0, -1 |
      +-      +-
>> polmat::coeff(B, s, 0)
      +-      +-
      | 0, 1 |
      |      |
      | 1, 1 |
      +-      +-

```

2.3 Basic arithmetic operations

We can use the standard arithmetical procedures of MuPAD for the matrix product, sum etc. Polmat comes with, for instance, the procedure for computing polynomial matrix greatest common divisor (GCD) which solves a system of equations

$$\begin{aligned} PA + QB &= D \\ RA + SB &= 0 \end{aligned}$$

where D is a right GCD and A, B are given polynomial matrices.

```
>> A:=polmat([[d-d^2, 2*d-d^2]], d):
>> B:=polmat([[d-1, d-1],[d-1, d-2]],d):
>> polmat::gcd(A, B, d)
      +-      +-
      | d - 1, d - 1 |
      |      |
      | 0, -1 |
      +-      +-

```

2.4 Linear equations

A standard theoretical and computational tool in linear control theory is linear Diophantine equation with polynomial matrices. Solving such the equation $AX + BY = C$, where

$$A = (d^2 + 1), B = (d, -d^2), C = (d)$$

for X and Y can be found in MuPAD as

```
>> [X, Y]:=polmat::axbyc(A, B, C, d)
--
|
| +-      +-
| | d - d t[1, 1] |
| +-      +-
--
+-      +-      +-
| 2          2 |
|- d + d t[2, 1] + t[1, 1] (d + 1) |
|
|          t[2, 1]
+-      +-

```

where t denotes an arbitrary polynomial matrix of size 2×1 that parametrises the solution set.

2.5 Spectral factorisation

A crucial computational step in various optimal control techniques is spectral factorisation of a polynomial matrix. It amounts to solving the following special quadratic equation with polynomial matrices

$$Z(s) = P^T(-s)JP(s) \quad (1)$$

where the constant matrix J consists of 1s and -1 s on the diagonal.

Consider the following polynomial matrix and compute its J-spectral factorisation

$$Z(s) = \begin{pmatrix} 0 & 1-s \\ 1+s & 1-s^2 \end{pmatrix}.$$

```
>> Z:=polmat([[0, 1-s],[1+s, 1-s^2]],s):
>> [P, J]:=polmat::spf(Z, s)
--+-      +-
| |          2 |
| |1/2 s + 1/2, - 1/4 s - 1/2 s + 5/4|
| |          |
| |          2 |
| |1/2 s + 1/2, - 1/4 s - 1/2 s - 3/4|
--+-      +-
      --
+-      +- |
| 1, 0 | |
|      | |
| 0, -1 | |
+-      +- |
      --

```

Note that this procedure does not necessarily return an exact solution. The algorithm is iterative and is stopped when adequate accuracy is reached.

However, every iteration is absolutely precise because it is computed by integer or rational arithmetic.

3. CONTROL DESIGN EXAMPLES

By means of a few examples we demonstrate the use of Polmat for solving the controller design by manipulations with polynomials and polynomial matrices.

Consider an inverted pendulum of Fig. 1. Suppose

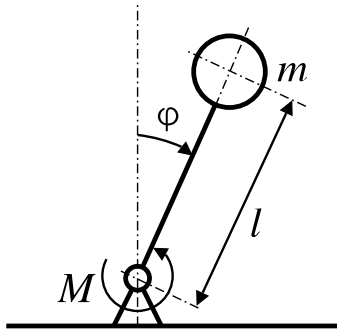


Fig. 1. Simple inverted pendulum

that friction in joint is negligible and all mass is concentrated in the ball. Then describe it by

$$m l^2 \frac{d^2 \varphi(t)}{dt^2} - m g l \sin \varphi(t) = -M(t), \quad (2)$$

where m is mass of the rod, l length of the rod, φ is angle and M is a torque.

Linearise the equation (2) about the point $\varphi(t) = 0$ rad and describe an inverted pendulum by transfer function

$$P(s) = \frac{1}{m g l - m l^2 s^2}$$

where input is the torque M and output is the angle φ . Choose $m = 0.5$ kg and $l = 1$ m and enter this system into MuPAD

```
>> P:=1/(m*l*g-m*l^2*s^2):
>> m:=1/2: l:=1: g:=981/100:
>> b:=polmat( numer(P), s):
>> a:=polmat( denom(P), s):
```

With these particular parameters the transfer function of the plant is

$$P(s) = \frac{b(s)}{a(s)} = \frac{200}{981 - 100 s^2}. \quad (3)$$

Evidently, the pendulum up is unstable position and the system (3) is not stable.

3.1 Feedback controller design via pole-placement

Design a controller via pole placement for our system (3). Consider the scheme of Fig. 2 and a

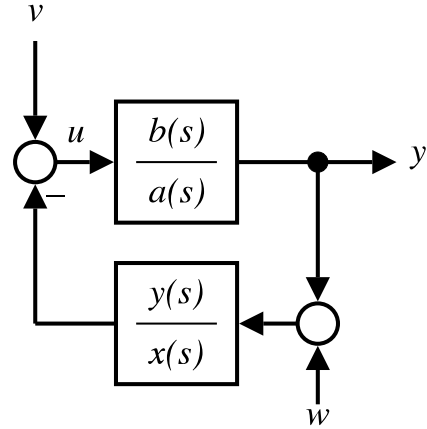


Fig. 2. Feedback control scheme controller having a form

$$C(s) = \frac{y(s)}{x(s)}.$$

Then the characteristic polynomial of a closed-loop system is

$$c(s) = a(s) x(s) + b(s) y(s). \quad (4)$$

To physically realise a controller, $c(s)$ must satisfy

$$\deg c(s) \geq 2 \deg a(s) - 1,$$

hence, the characteristic polynomial for the system (3) has the form

$$c(s) = (s^2 + 2 \zeta \omega_n s + \omega_n^2) (s + c_1).$$

Choose

$$\omega_n = 20 \sqrt{2}, \quad \zeta = \frac{\sqrt{2}}{2}, \quad c_1 = 100,$$

corresponding characteristic polynomial is

$$c(s) = (s^2 + 40 s + 800) (s + 100),$$

and solve (4) for $x(s)$ and $y(s)$.

```
>> c:=polmat((s^2+40*s+800)*(s+100), s):
>> [x, y]:=polmat::axbyc(a, b, c, s,
"y-minimal"):
>> C:=polmat::tf(y, x):

- 480981 s - 8137340
-----
200 s + 28000
```

The resulting controller is

$$C(s) = \frac{-480981 s - 8137340}{200 s + 28000}.$$

Zero and pole map and step response of closed-loop system is in Fig. 3.

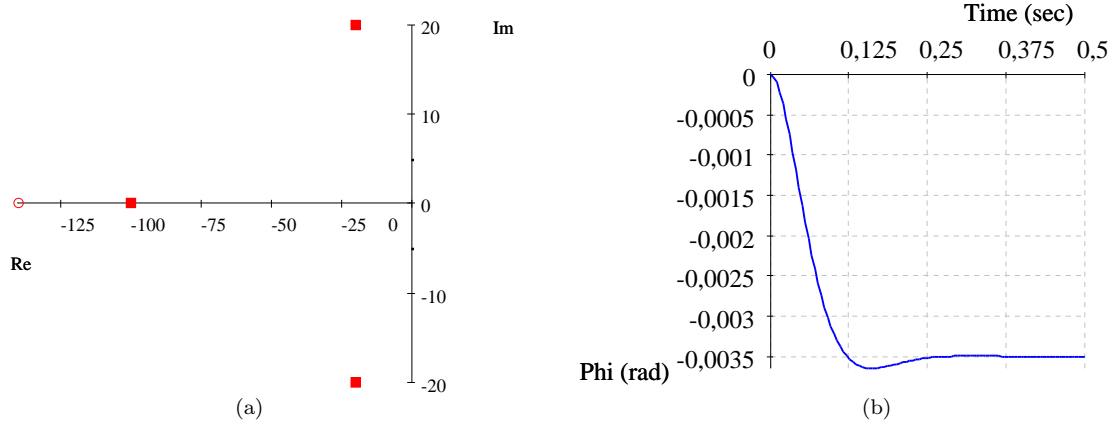


Fig. 3. Zero and pole map (a) and step response (b) of closed-loop system

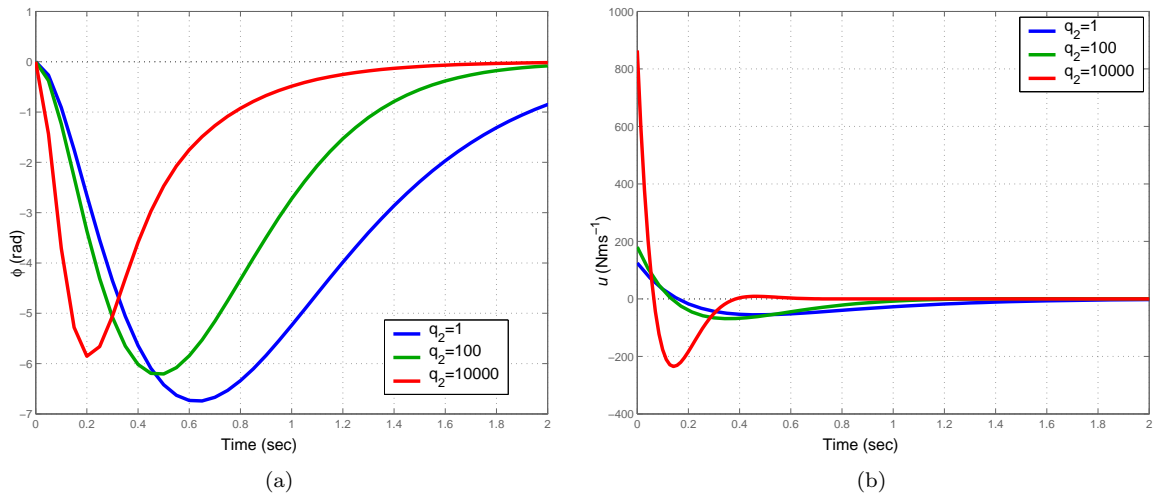


Fig. 4. Impulse response of closed-loop system, output (a), manipulated variable (b) for various weights q_2 (impulse is applied to input w)

3.2 Feedback controller design via LQG

Our objective is to design a controller

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

for the system (3) which minimises the standard quadratic optimisation criterion

$$\lim_{t \rightarrow \infty} \int_0^t \{u^T(t) r_2 u(t) + y^T(t) q_2 y(t)\} dt$$

where $u(t)$ is the control input and $y(t)$ is the measured output. Let $v(t)$ be the disturbing torque acting on the pendulum with covariance r_1 and $w(t)$ is a noise that adds to measured angular position with covariance q_1 , see Fig. 2. We consider the covariances of the input noise and output noise

$$q_1 = 5 \cdot 10^{-6}, \quad r_1 = 5 \cdot 10^{-6}$$

and weights on the output (controlled) signals and the input (control) signal

$$q_2 = 1, \quad r_2 = 1.$$

```
>> q1:=r1:=polmat(5/1000000, s):
>> q2:=r2:=polmat(1, s):
```

A controller design consists of three steps: solving two spectral factorisations and one linear (Diophantine) equation:

$$a(s) r_1 a^*(s) + b(s) q_1 b^*(s) = f(s) f^*(s)$$

$$a(s) r_2 a^*(s) + b^*(s) q_2 b(s) = g(s) g^*(s)$$

$$a(s) x(s) + b(s) y(s) = f(s) g(s),$$

where $X^*(s) = X^T(-s)$ and taking the solution with $\deg y(s) < \deg a(s)$.

```
>> f:=polmat::spf(
a*r1*polmat::conjugatetranspose(a,s)+
b*q1*polmat::conjugatetranspose(b,s),
s)[1]:
>> g:=polmat::spf(
a*r2*polmat::conjugatetranspose(a,s)+
polmat::conjugatetranspose(b,s)*q2*b,
s)[1]:
```

```

>> c:=polmat(f*g, s):
>> [x, y]:=polmat::axbyc(a, b, c, s,
                        "degy<dega"):
>> subs([x, y], t=0):
>> C:=simplify(factor(
                        1/polmat::tf(op(%)))):
>> float(C)

      0.5 (5581418.223 s + 17481743.91)
-----
                          2
-281579.8149s-22360.67977s -1553557.246

```

The controller is

$$C(s) = \frac{2790709.112 s + 8740871.957}{-22360.68 s^2 - 281579.81 s - 1553557.22}$$

Fig. 4 shows impulse responses for various q_2 .

4. CONCLUSIONS

This paper presented a new package for computing with polynomial matrices. The package takes advantage of symbolic computation performance of MuPAD and thus provides a useful alternative to the few existing numerical libraries for polynomial matrices. The Polmat library is available for free download as well as the MuPAD system, which makes the computation with polynomial matrices accessible to everyone. The development of the Polmat package is going on in near future.

ACKNOWLEDGEMENTS

The authors would like to thank to developers of MuPAD for making it accessible to individual students and researchers.

REFERENCES

- Augusta, P. (2005). Polmat: MuPAD library for symbolic computation with polynomial matrices. Master's thesis. Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University. Prague.
- Augusta, P. and Zdeněk Hurák (2005). *Polmat: MuPAD library for symbolic computation with polynomial matrices*. Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University. Prague, Czech Republic. <http://polmat.wz.cz>.
- Benner, P., V. Mehrmann, V. Sima, S. Van Huffel and A. Varga (1997). *SLICOT - A Subroutine Library in Systems and Control Theory*. NICONET Network. <http://www.win.tue.nl/wgs/slicot.html>.

- Grimble, M.J. and Kučera, V., Eds.) (1996). *Polynomial methods for control system design*. Springer-Verlag London.
- Halmo, L. (2004). *PolPack++: C++ library for computing with polynomial matrices*. Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University. Prague, Czech Republic. <http://polpackplusplus.sourceforge.net>.
- Hunt, K.J., Ed.) (1993). *Polynomial methods in optimal control and filtering*. Peter Peregrinus.
- INRIA and ENPC (2000). *Scilab*. Scilab Consortium. <http://scilabsoft.inria.fr/>.
- Kailath, T. (1980). *Linear Systems*. Prentice Hall.
- Kučera, V. (1979). *Discrete Linear Control*. John Wiley and Sons.
- Kwakernaak, H. and M. Šebek (1998). *Polynomial toolbox for Matlab*. PolyX, Ltd.. Prague, Czech Republic. <http://www.polyx.com>.
- Paděra, M. (2004). *Java package for computing with polynomial matrices*. Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University. Prague, Czech Republic. <http://klokansh.cvut.cz/~padera/polynomial>.
- SciFace (2000). *MuPAD*. SciFace. <http://www.mupad.com/>.
- Štefko, P. (2004). *C library for computing with polynomial matrices for the TI-89/92 programmable calculators*. Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University. Prague, Czech Republic. <http://ptoolti89.wz.cz>.