VYUŽITÍ POLYNOMIÁLNÍHO TOOLBOXU™ 2.0 PRO MATLAB ™ VE VÝUCE NA FEL ČVUT

Ing. Martin Hromčík

Ústav teorie informace a automatizace, Akademie věd České republiky,
Pod vodárenskou věží 4, 18208 Praha 8.
Tel: +420-2-6605 2240; Fax: +420-2-688 4554
E-mail: hromcik@utia.cas.cz

Trnkova laboratoř automatického řízení, České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra řídící techniky, Karlovo náměstí 13,
121 35 Praha 2.
Tel: +420-2-2435 7277; Fax +420-2-2435 7266

Abstrakt: Tato zpráva obsahuje shrnutí prvních zkušeností autora s využíváním Polynomiálního Toolboxu™ verze 2.0 pro Matlab™ [1,2] při výuce polynomiálních metod návrhu řídících obvodů ve cvičeních předmětu Algebraické metody syntézy, vyučovaném Katedrou řídící techniky Fakulty elektrotechnické ČVUT v Praze pro 5. ročník oboru Technická kybernetika inženýrského studia[1].

1 Úvod

Polynomiální či algebraické metody návrhu řídících obvodů tvoří vedle klasických frekvenčních metod a metod stavového prostoru třetí základní nástroj pro teoretický návrh lineárních regulačních obvodů. Jejich vznik je datován do 70 let a od svého vzniku prošly mohutným vývojem [7,8]. Jejich výhodou oproti stavovým metodám bývá přímočaré řešení problému vycházející pouze ze znalosti vstupně-výstupního modelu bez potřeby zavádět a odhadovat další pomocné stavové veličiny. Výsledky lze často rovněž zajímavě interpretovat a objasňovat tak na první pohled ne zcela zřejmé souvislosti mezi zdánlivě rozdílnými úlohami řízení.

Polynomiální metody jsou dnes standardně zařazovány do výuky oborů řídící techniky a automatizace na technických vysokých školách ve formě specializovaných kurzů či v rámci jiných odborných předmětů věnujících se problematice teoretického návrhu řídících systémů. Literatura věnující se této problematice je dnes dobře dostupná a nabídka široká, ovšem snahy o zařazení komplexních úloh syntézy řízení do cvičení byly donedávna těžko realizovatelné pro nedostatek vhodných výpočetních
programů pro práci s polynomy a polynomiálními maticemi. Veškeré výpočty pak bylo nutné provádět „ručně“ na papíře či komplikovaným převodem úlohy do stavového prostoru což hodnotu polynomiálního přístupu poněkud snižovalo.

V posledních letech však byl v oblasti numerických algoritmů pro operace s polynomiálními maticemi udělán velký pokrok. Byla publikována série článků pojednávajících o nových výpočetních postupech s důrazem na jejich numerickou stabilitu nezbytnou pro spolehlivou implementaci na číslicovém počítači [9,10,11]. Praktickým důsledkem těchto snah je pak *Polynomiální toolbox™ pro Matlab™,* první prakticky použitelný software pro operace s polynomy a polynomiálními maticemi nezbytné pro návrhu řídících systémů algebraickými metodami [1,2,3].

[1] POLYNOMIÁLNÍ TOOLBOX™ je ochranná známka firmy PolyX, s.r.o. MATLAB™, SIMULINK™ A CONTROL SYSTEMS TOOLBOX™ jsou registrované ochranné známky firmy The Mathworks Inc.

2 Polynomiální toolbox pro Matlab

Polynomiální toolbox™ pro Matlab™ je programový balík pro operace s polynomy a polynomiálními maticemi vhodný pro praktický návrhu řídících systémů algebraickými metodami. Skládá se z „m-souborů“ interpretovatelných výpočetním programem *Matlab™*.

Verze 1.5 a 1.6 tohoto nástroje, určené pro verzi 4.2 *Matlabu™* a vyšší, byly volně k dispozici na internetu ke stažení jako freeware [3]. Na naší katedře byla tato verze v kombinaci s Matlabem 4.2c používána minulé dva roky.

Koncem loňského roku se pak objevila na trhu zcela nová verze 2.0 [1,2] založená na novém Matlabu 5 využívající všech jeho nových vlastností. Narozdíl od starších verzí toolboxu je nová verze objektově orientovaná a takřka všechny standardní funkce a operace byly pro objekt polynomiální matice předefinovány. Díky této strategii je tak nový toolbox mnohem uživatelsky přívětivější – například zadání polynomu, zformování polynomiální matice či součet dvou takových matic se provádí zcela přirozeně s použitím standardní matlabské notace:

```
» p1 = s^2 + s + 1

p1 =
     1 + s + s^2

» p2 = (s+1)^3

p2 =
     1 + 3s + 3s^2 + s^3

» P = [p1 p2]


P =
      1 + s + s^2     1 + 3s + 3s^2 + s^3

» P2 = P + [p1 p1]

P2 =
      2 + 2s + 2s^2     2 + 4s + 4s^2 + s^3
```

Pro zadávání rozsáhlejších polynomiálních matic je k dispozici grafický editor.

Nová verze je rovněž rychlejší a spolehlivější díky novým algoritmům, založeným například na Sylvestrovských [11] metodách či rychlé Fourierově transformaci (FFT) [12], které byly nově vyvinuty.

Co se týká návrhu řídících systémů, Polynomiální toolbox obsahuje nástroje potřebné pro všechny základní i méně obvyklé úlohy syntézy. Systémy spojité i s diskrétním časem lze s využitím toolboxu modelovat polynomiálními maticovými zlomky. K dispozici jsou funkce na řešení lineárních rovnic s polynomy a polynomiálními maticemi (diofantické rovnice, symetrické rovnice, symetrické bilaterální rovnice, ...), kvadratických polynomiálních rovnice (spektrální faktorizace, J-spektrální faktorizace, ...), aj. Pro řadu standardních úloh (například umístění pólů, deadbeat, H2 a H∞ optimalizace aj.) jsou navíc připraveny speciální programy. Celá řada maker je pak rovněž věnována technikám robustního řízení (pro systémy s jedním parametrem, intervalové a polytopické polynomy, ...). Kromě toho lze všechny výsledky syntézy okamžitě testovat s využitím simulačního nástroje *Simulink™* [5] - příslušný blok pro definování lineárního řídícího systému ve formě polynomiálního (maticového) zlomku je rovněž součástí toolboxu.

3 Využití toolboxu ve výuce

Tyto možnosti Polynomiálního toolboxu lze s výhodou využít ve cvičeních předmětů věnujících se algebraickým metodám syntézy. Studenti tak mohou věnovat mnohem více času otázkám souvisejícím s kvalitou navrhovaného řízení a ladění řídícího systému než mechanickému numerickému řešení již zformulované úlohy. Jako velmi užitečná se jeví rovněž možnost spolupráce toolboxu se Simulinkem: studenti tak ihned vidí chování výsledné smyčky, mohou jednoduše zkoumat vliv změn parametrů či počátečních podmínek soustavy, odolnost smyčky vůči šumům, vliv nelinearit soustavy apod. Cvičení se tak stává více praktickým a pro studenty užitečnějším a zajímavějším. Důležitá je rovněž možnost odpovědného porovnání polynomiálního a stavového přístupu při řešení praktických úloh návrhu řízení na počítači s využitím Polynomiálního toolboxu v prvém případě a *Control Systems Toolboxu™* [6] pro *Matlab™* v případě stavových metod. Právě tak mohou vyniknout přednosti polynomiálního přístupu kdy vycházíme pouze ze vstupně-výstupních informací o soustavě bez potřeby explicitně zavádět a odhadovat všechny stavy soustavy.

Ve cvičeních předmětu Algebraické metody syntézy byl *Polynomiální toolbox* využíván při řešení těchto úloh syntézy pro SISO systémy s využitím jmenovaných funkcí:

- **Umístění pólů (pole-placement)**

  - řešení diofantické rovnice (funkce **`axbyc`** `[2]`),
  - přímo pomocí funkce **`pplace`** `[2]`.

- **Časově optimální diskrétní řízení (deadbeat)**

  - jako umístění pólů do počátku souřadnic (viz výše),
  - přímo (funkce **`debe`** `[2]`).

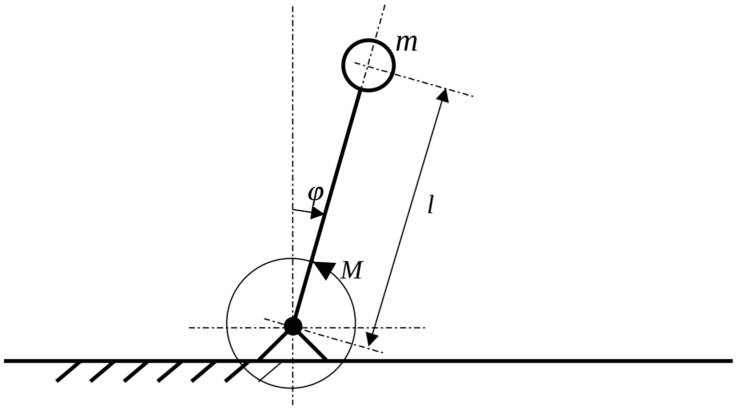
- **H2 optimální řízení (LQG)**

  - 2× spektrální faktorizace (funkce **`spf`** `[2]`) + řešení diofantické rovnice (umístění pólů),
  - přímo – funkce **`plqg`** `[2].`

4 Příklad použití toolboxu při návrhu řízení

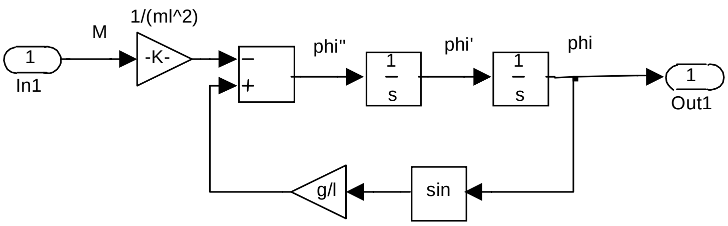
Jako velmi jednoduchý příklad použití toolboxu při návrhu řízení uvádím řešení problému stabilizace inverzního kyvadla dynamickou výstupní zpětnou vazbou algebraickými metodami s pomocí *Polynomiálního toolboxu*.

Schematický náčrt soustavy s vyznačením potřebných veličin je na následujícím obrázku:



Obr. 1: Inverzní kyvadlo

Za předpokladu, že veškerá hmotnost kyvadla je soustředěna v kouli a že tření v kloubu je zanedbatelné, můžeme tuto soustavu popsat následující (nelineární) diferenciální rovnicí:
Příslušné schema sestavené v simulačním nástroji *Simulink™* vypadá pak následovně:



Obr. 2: Simulinkový model inverzního kyvadla

Vytvořit linearizovaný spojitý i vzorkovaný vstupně-výstupní model (přenos) této soustavy je s využitím Matlabu [3], Polynomiálního toolboxu [1,2] a Control Systems Toolboxu [5] otázkou pár řádků kódu:

```
» [a,b,c,d] = linmod('invpend');  %‘invpend.mdl’ je jméno
                                  % Simulinkového modelu výše

» sys_ct = ss(a,b,c,d);

» [n_ct,d_ct] = lti2lmf(sys_ct)   % SPOJITÝ model

Constant polynomial matrix: 1-by-1

n_ct =                            % čitatel přenosu

     1

d_ct =                       % jmenovatel přenosu

     10 - s^2


» sys_samp = c2d(sys_ct, 0.5);
```

```
» [n_samp, d_samp] = lti2lmf(sys_samp) % DISKRETIZOVANÝ model

n_samp =                          % čitatel přenosu

    -0.21 - 0.21z

d_samp =                          % jmenovatel přenosu

     1.4 - 6.9z + 1.4z^2
```

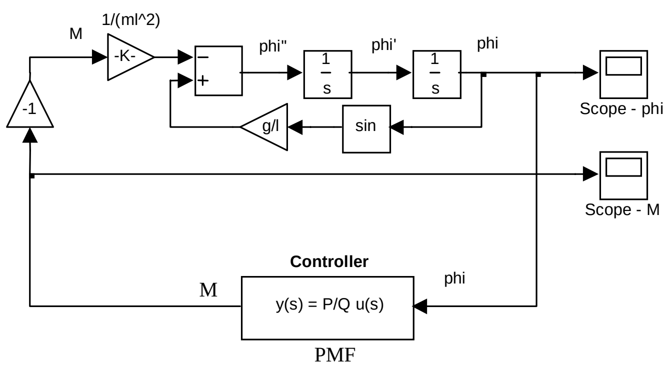
Pro návrh spojitého regulátoru, který umístí póly výsledné smyčky do (stabilní) polohy -1, -1+j a -1-j, můžeme sestavit polynom $c(s)=(s+1)(s+1+j)(s+1-j)$a vyřešit příslušnou diofantickou rovnici [7,8]:

```
» [n_cont, d_cont] = axbyc(n_ct, d_ct, (s+1)*(s+1+j)*(s+1-j))
```

nebo přímo zavolat makro **pplace** [2] s příslušnými argumenty:

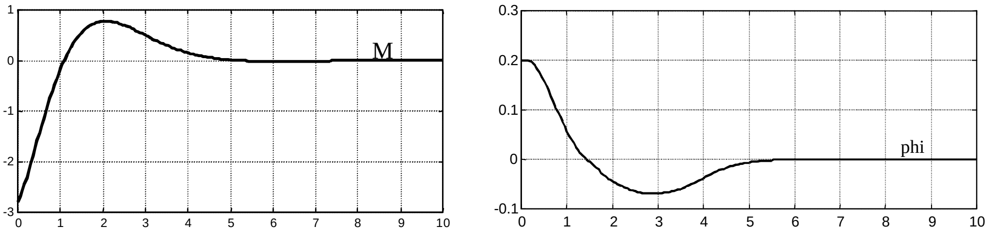
```
» [n_cont, d_cont] = pplace(n_ct, d_ct, [-1,-1+j, -1-j])

n_cont =        % čitatel přenosu regulátoru

     32 + 14s

d_cont =        % jmenovatel přenosu regulátoru

    -3 – s
```

Zda tento řídící systém, navržený na základě lineárního modelu, skutečně ustabilizuje náš nelineární systém, snadno ověříme pro různé počáteční výchylky kyvadla pomocí Simulinku – model z obr. 2 doplníme o regulátor (zde využijeme simulinkový blok z Polynomiálního toolboxu [1,2]) a uzavřeme smyčku:



Obr. 3: Simulinkový model uzavřené smyčky

Výsledky simulace pro $\varphi(0) = 0.2$ jsou na následujícím obrázku:



Obr. 4: Výsledky simulace pro $\varphi(0) = 0.2$ (spojitý regulátor)

Stejným způsobem můžeme navrhnout regulátor číslicový (diskrétní), například deadbeat [7,8]. Vyjdeme z diskretizovaného lineárního modelu. Výsledek dostaneme jedním ze tří následujících příkazů Polynomiálního toolboxu.

```
» [n_cont, d_cont] = axbyc(n_samp, d_samp, z^3);
```

(přiřazení charakteristického polynomu z3)

```
» [n_cont, d_cont] = pplace(n_samp, d_samp, 0);
```

(přesunutí pólů do počátku)

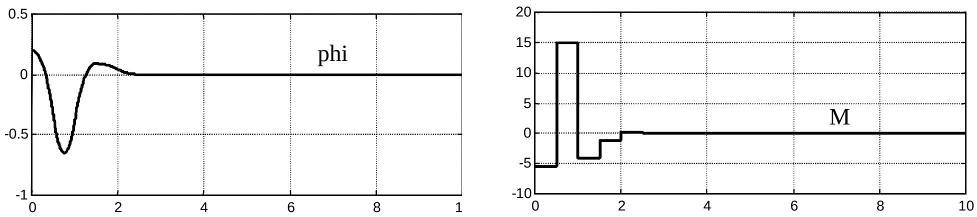
```
» [n_cont, d_cont] = debe(n_samp, d_samp)
```

(přímo pomocí makra **debe** [2])

Výsledek má tvar:

```
» [n_cont, d_cont] = debe(n_samp, d_samp)

n_cont =        % ČITATEL přenosu regulátoru
     4.1 - 20z

d_cont =        % JMENOVATEL přenosu regulátoru
     0.63 + 0.74z
```

Chování regulátoru lze opět ověřit s využitím Simulinkového schematu na obr. 2.



Obr. 5: Výsledky simulace pro $\varphi(0) = 0.2$ (deadbeatový regulátor)

Poděkování

Tato práce vznikla za podpory Grantové agentury České republiky (grant 102/99/1368) a MŠMT České republiky (projekt č. VS97/034).

Literatura

[1] THE POLYX, Ltd.: PolyX Home Page, http://www.polyx.cz, http://www.polyx.com.
[2] THE POLYX, Ltd.: Polynomial Toolbox 2.0 Manual, The PolyX, Prague, 1999.
[3] ŠEBEK, M. – KWAKERNAAK, H.: Polynomial Toolbox Home Page, http://www.math.utwente.nl/polbox.
[4] THE MATHWORKS Inc.: Using MATLAB, Version 5, The MathWorks Inc., 1997.
[5] THE MATHWORKS Inc.: Using SIMULINK, Version 2, The MathWorks Inc., 1997.
[6] THE MATHWORKS Inc.: Control Systems Toolbox, User's Guide, The MathWorks Inc., 1996.
[7] KUČERA, V: Discrete Linear Control: The Polynomial Equation Approach, Academia Praha, 1979.
[8] KUČERA, V.: Analysis and Design of Discrete Linear Control Systems, Academia Praha, 1991.
[9] HENRION, D: Reliable Algorithms for Polynomial Matrices, kandidátská disertace, UTIA AVČR, Praha, 1998.
[10] Henrion, D. – Šebek, M.: Symmetric Matrix Polynomial Equation: Interpolation Results, Automatica, Vol. 34, No. 7, pp. 811-824, 1998.
[11] Henrion, D. – Šebek, M: Efficient Algorithms for Polynomial Matrix Triangularization, IEEE Transactions on Automatic Control, April 1999.
[12] Hromčík, M. – Šebek, M.: New Algorithm for Polynomial Matrix Determinant Based on FFT, European Control Conference, Karlsruhe, 31 August – 3 September 1999.