



Stavové řízení

Tato publikace vznikla jako součást projektu CZ.04.1.03/3.2.15.2/0285 „Inovace VŠ oborů strojního zaměření“, který je spolufinancován evropským sociálním fondem a státním rozpočtem České republiky

Milan Turek, 11. 6. 2007

1 Obsah

2	Stavový model	3
3	Pozorovatel	3
4	LQ design.....	5
5	Stavové řízení s integrátorem na vstupu	8
6	Stavové prediktivní řízení	11

Metod pro návrh řízení lineárních systémů existuje velké množství od asi nejznámější metody Ziegler-Nicholsovy až po metody návrhu robustního řízení (H-infinity, μ syntéza).

Dále uvedené metody stavového návrhu řízení [1] se vyznačují dobře podloženou teorií a dobrými výsledky. Vycházejí ze stavového modelu řízeného systému a výsledný regulátor počítá akční zásah na základě stavů systému. Stav systému (např. poloha, rychlost, proud protékající vedením) nelze ve většině případů měřit přímo nebo nemají fyzikální ekvivalent, proto se k určení jejich hodnot používá pozorovatel.

2 Stavový model

Stavový model spojitého systému popisuje chování systému pomocí závislosti derivace stavů na jejich aktuální hodnotě. Je dán následujícími maticovými rovnicemi.

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}\end{aligned}\tag{1}$$

kde \mathbf{x} je vektor stavů, \mathbf{u} je vektor vstupů, \mathbf{y} je vektor výstupů, matice \mathbf{A} popisuje závislost derivací stavů na aktuálních hodnotách stavů, matice \mathbf{B} popisuje závislost derivací stavů na vstupech, matice \mathbf{C} popisuje závislost výstupů na aktuálních hodnotách stavů a matice \mathbf{D} popisuje závislost výstupů na vstupech.

Stavový model diskrétního systému popisuje chování systému pomocí závislosti hodnot stavů v následujícím časovém okamžiku na jejich hodnotě v aktuálním časovém okamžiku. Následující maticové rovnice popisující stavový model diskrétního systému mají stejnou strukturu jako u stavového modelu spojitého systému. Význam jednotlivých symbolů je obdobný jako u spojitého systému – $\mathbf{x}[k + 1]$ je vektor stavů v následujícím časovém okamžiku, $\mathbf{x}[k]$ je vektor stavů v aktuálním časovém okamžiku, $\mathbf{u}[k]$ je vektor vstupů, $\mathbf{y}[k]$ je vektor výstupů, matice \mathbf{A} popisuje závislost hodnot stavů v následujícím časovém okamžiku na aktuálních hodnotách stavů, matice \mathbf{B} popisuje závislost hodnot stavů v následujícím časovém okamžiku na vstupech, matice \mathbf{C} popisuje závislost výstupů na aktuálních hodnotách stavů a matice \mathbf{D} popisuje závislost výstupů na vstupech.

$$\begin{aligned}\mathbf{x}[k + 1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] \\ \mathbf{y}[k] &= \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k]\end{aligned}\tag{2}$$

3 Pozorovatel

Pozorovatel slouží k určení aktuálních hodnot stavů řízeného systému, jinak řečeno k pozorování jeho chování. Asi nejznámějším typem pozorovatele je Kalmanův filtr, který je schopen pracovat i při vysoké úrovni šumu. Pro běžné účely ale dostačuje mnohem jednodušší typ pozorovatele. Navíc v současnosti, kdy drtivá většina řídicích systémů je diskrétní, má smysl pracovat pouze

s diskrétním pozorovatelem. Proto je dále uvažována pouze diskrétní verze pozorovatele, i když lze analogickým způsobem navrhnout spojitou verzi.

Pozorovatel je tvořen modelem systému, do kterého vstupuje stejný řídicí signál jako do reálného systému, tzv. známý vstup. V ideálním případě by takováto implementace dostačovala, abychom získali hodnoty stavů. V reálných případech ale na systém působí další neměřitelné vlivy, tzv. poruchy a také model přesně neodpovídá reálnému chování systému. Tyto vlivy musí být kompenzovány. Kompenzace se provádí přičtením vážené hodnoty rozdílu měřeného výstupu systému a výstupu pozorovatele k hodnotám stavů. Vektor vah \mathbb{H} se nazývá matice pozorovatele. Pozorovatel je popsán následujícími maticovými rovnicemi.

$$\begin{aligned}\hat{\mathbf{x}}[k + 1] &= \mathbf{A}\hat{\mathbf{x}}[k] + \mathbb{H}(\mathbf{y}[k] - \hat{\mathbf{y}}[k]) + \mathbb{B}\mathbf{u}[k] \\ \hat{\mathbf{y}}[k] &= \mathbf{C}\hat{\mathbf{x}}[k] + \mathbb{D}\mathbf{u}[k]\end{aligned}\quad (3)$$

Dosazením za odhad výstupu systému a úpravou se získá

$$\begin{aligned}\hat{\mathbf{x}}[k + 1] &= (\mathbf{A} - \mathbb{H}\mathbf{C})\hat{\mathbf{x}}[k] + \mathbb{H}\mathbf{y}[k] + (\mathbb{B} - \mathbb{H}\mathbb{D})\mathbf{u}[k] \\ \hat{\mathbf{y}}[k] &= \mathbf{C}\hat{\mathbf{x}}[k] + \mathbb{D}\mathbf{u}[k]\end{aligned}\quad (4)$$

kde $\hat{\mathbf{x}}[k]$ je odhad stavů systému, $\hat{\mathbf{y}}[k]$ je odhad výstupu systému, $\mathbf{y}[k]$ je měřený výstup systému a $\mathbf{u}[k]$ je známý vstup do systému. Matice \mathbf{A} , \mathbf{B} , \mathbf{C} a \mathbf{D} jsou parametry modelu systému (viz. Stavový model).

Návrh pozorovatele tedy spočívá v určení hodnot matice pozorovatele \mathbb{H} . Z požadavků, aby byl pozorovatel stabilní a zároveň reagoval rychleji než pozorovaný systém, vyplývá, že póly pozorovatele musí ležet uvnitř jednotkové kružnice a musí být blíže k nule než póly pozorovaného systému. Vzhledem k tomu, že póly pozorovatele odpovídají kořenům matice $(\mathbf{A} - \mathbb{H}\mathbf{C})$, lze matici pozorovatele navrhnout tak, že se zvolí vhodné póly pozorovatele a relativně jednoduchým výpočtem se dopočítají koeficienty matice pozorovatele. Tento výpočet lze provést automatizovaně pomocí vhodného softwaru, např. Matlab nebo NI LabView.

Tabulka 1: Příklad návrhu pozorovatele v prostředí Matlab

```
% návrh pozorovatele pro závaží na pružině na které působí říditelná síla
% systém je popsán rovnicí: mx'' + bx' + kx = F
m = 0.2;      % hmotnost
b = 0.2;      % tlumení
k = 1;        % tuhost

T = 0.01;     % perioda vzorkování

% sestavení matic popisu systému (spojitá verze)
A = [0 1; -k/m -b/m];
B = [0; 1/m];
C = [1 0];
```

```

D = 0;

sys = ss(A,B,C,D); % spojitý model

% kontrola pozorovatelnosti
n = size(B, 1); % n řád řízeného systému
Mo = obsv(sys); % výpočet matice pozorovatelnosti

if (rank(Mo) ~= n)
    error('System není pozorovatelný.');
```

```

end;

% převod spojitého modelu na diskrétní se vzorkovací periodou T
sys_d = c2d(sys,T);

% výpis pólů diskrétního modelu
pole(sys_d)

% póly mají hodnoty 0.9948 + 0.0217i a 0.9948 - 0.0217i
% zvolí se póly pozorovatele tak aby byly rychlejší (tj. blíže nule)
op = [0.8 0.9];

% spočítat a vypsát matici pozorovatele
H = place(sys_d.A',sys_d.C',op) '

%výsledná matice pozorovatele je sloupcový vektor s koeficienty
% 0.2896 a 1.6634
```

4 LQ design

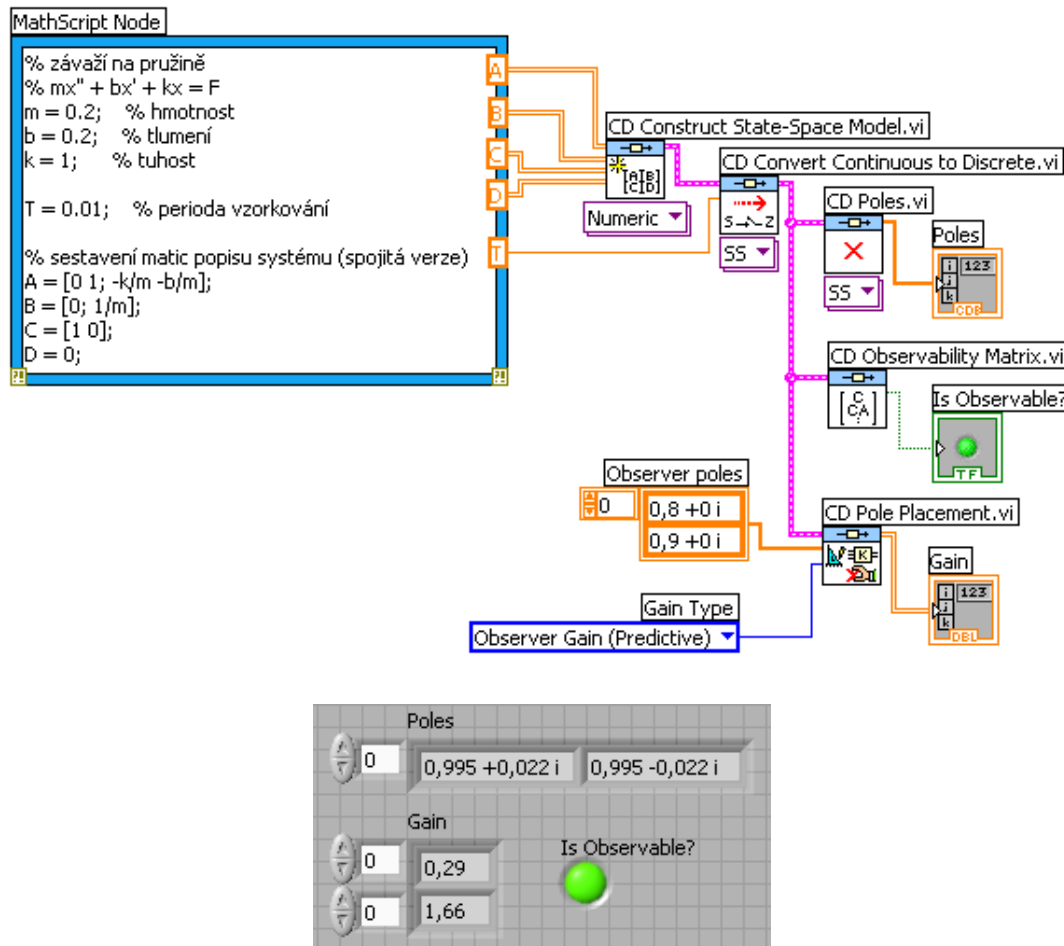
LQ design je nejrozšířenější metodou stavového návrhu řízení. Velice často se používá ve spojení s kalmanovým filtrem, který plní úlohy pozorovatele a filtru odstraňujícího šum z naměřených hodnot (tzv. LQG design). LQ design umožňuje definovat zda jsou důležitější malé hodnoty zásahů regulátoru nebo rychlá reakce řízeného systému.

Akční zásah regulátoru je počítán jako skalární součin vektoru zesílení regulátoru a stavového vektoru řízeného systému. Návrh řízení tedy spočívá v nalezení hodnot vektoru zesílení regulátoru takových, aby minimalizovaly cenovou funkci

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + 2\mathbf{x}^T \mathbf{N} \mathbf{u} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt, \quad (5)$$

kde \mathbf{x} je vektor stavů systému, \mathbf{u} je vektor vstupů systému a \mathbf{Q} , \mathbf{N} a \mathbf{R} jsou volené matice koeficientů. Čím větší je hodnota koeficientu, tím více je hodnota vstupu/stavu navrženým regulátorem minimalizována. Například pokud je třeba dosáhnout co nejrychleji co nejmenší hodnoty prvního stavu a na hodnotách ostatních stavů a vstupů nezáleží, je třeba nastavit hodnotu v prvním řádku a prvním sloupci matice \mathbf{Q} na vysokou hodnotu a ostatní hodnoty koeficientů

ponechat malé. Návrh koeficientů je nejobtížnější částí návrhu LQ regulátoru a ve většině případech je při jejich návrhu třeba postupovat metodou pokusu a omylu tak dlouho, dokud nejsou splněny požadavky na řízení.



Obrázek 1: Příklad návrhu pozorovatele v prostředí NI LabView

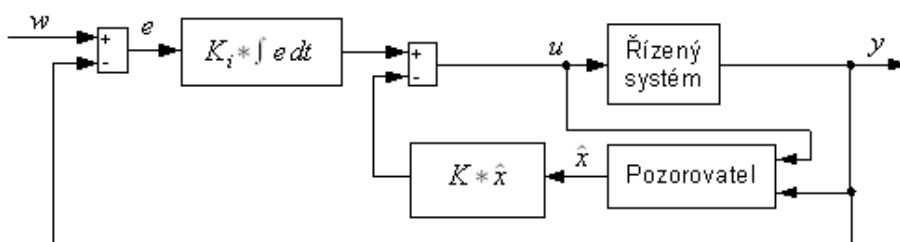
Takto navržený regulátor je vzhledem ke stavům systému proporcionální, ale v praxi je pro kompenzaci konstantního zatížení třeba, aby měl regulátor i integrační složku. Návrh zesílení integrační složky je v případě LQ designu velice jednoduchý. Stačí do modelu systému přidat mezi stavy integrál výstupu systému a hodnota odpovídající integračnímu stavu v navrženém vektoru zesílení je poté zesílením integrátoru. Integrátor se následně zapojí paralelně ke stavovému regulátoru.

Přidání integračního stavu se provede následovně. Hodnota integračního stavu x_i je rovna hodnotě integrálu výstupu

$$x_i = \int y dt = \int (Cx + Du) dt. \quad (6)$$

Derivace tohoto stavu je tedy rovna výstupu. Rozšířením rovnic stavového popisu 1 se získá model systému, který mezi stavy obsahuje i integrál výstupu.

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_i \end{bmatrix} &= \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} + \begin{bmatrix} B \\ D \end{bmatrix} u \\ y &= [C \quad 0] \begin{bmatrix} x \\ x_i \end{bmatrix} + Du \end{aligned} \quad (7)$$



Obrázek 2: Struktura řízení pomocí LQ regulátoru s připojenou integrační složkou

Zesílení regulátoru lze touto metodou spočítat jak pro spojité tak i diskrétní řízení (převedením spojitého stavového popisu na ekvivalentní diskrétní). Odvození minima cenové funkce pro složitější systémy není triviální úlohou, ale opět lze využít software Matlab a LabView, které mají tuto metodu návrhu řízení zabudovanou ve svých rozšířeních.

Tabulka 2: Příklad návrhu LQ regulátoru pomocí Matlabu

```
% závaží na pružině
% mx'' + bx' + kx = F
m = 0.2; % hmotnost
b = 0.2; % tlumení
k = 1; % tuhost

T = 0.01; % perioda vzorkování

% sestavení matic popisu systému (spojitá verze)
A = [0 1; -k/m -b/m];
B = [0; 1/m];
C = [1 0];
D = 0;

n = size(B, 1); % n řád řízeného systému
m = size(B, 2); % m počet vstupních veličin

% do stavů doplnit integrál výstupu (předpoklad jednoho výstupu)
Ai = [A zeros(n,1); C 0];
Bi = [B; D];
```

```

Ci = [C 0];

sys = ss(Ai,Bi,Ci,D); % spojitý model

% kontrola řiditelnosti
Mc = ctrb(sys);
ni = size(Bi, 1); % ni řád řízeného systému s integrálem výstupu
if (rank(Mc) ~= ni)
    error('System není řiditelný.');
```

```

end;

% volba matic koeficientů cenové funkce Q a R
% (N není nutné definovat. Pokud se nezadá je považována za nulovou.)
% Pozn.: Nejvíce chceme minimalizovat odchylku polohy (první stav),
%       na rychlosti a integrálu polohy nám nezáleží (druhý a třetí
%       stav), ale chceme také malé akční zásahy.
Q = eye(ni,ni); % matice Q musí být čtvercová se stejným řádem jako systém
Q(1,1) = 1e3; % chceme nejvíce minimalizovat první stav (polohu)

R = eye(m); % matice R musí být čtvercová se stejným řádem jako počet
           % vstupů
R = R*10; % přidat na významu minimalizaci akčních zásahů

% návrh diskrétního regulátoru (vektoru zesílení) pro spojitý systém
K = lqrd(Ai, Bi, Q, R, T);

% z vektoru zesílení získat zesílení integrátoru
% (zesílení posledního stavu - integrálu polohy)
% (předpoklad jednoho výstupu)
Ki = K(ni) % Ki = 0.3027

% z vektoru zesílení odstranit zesílení integrátoru
K = K(1:n) % K = [8.6810 1.7063]
```

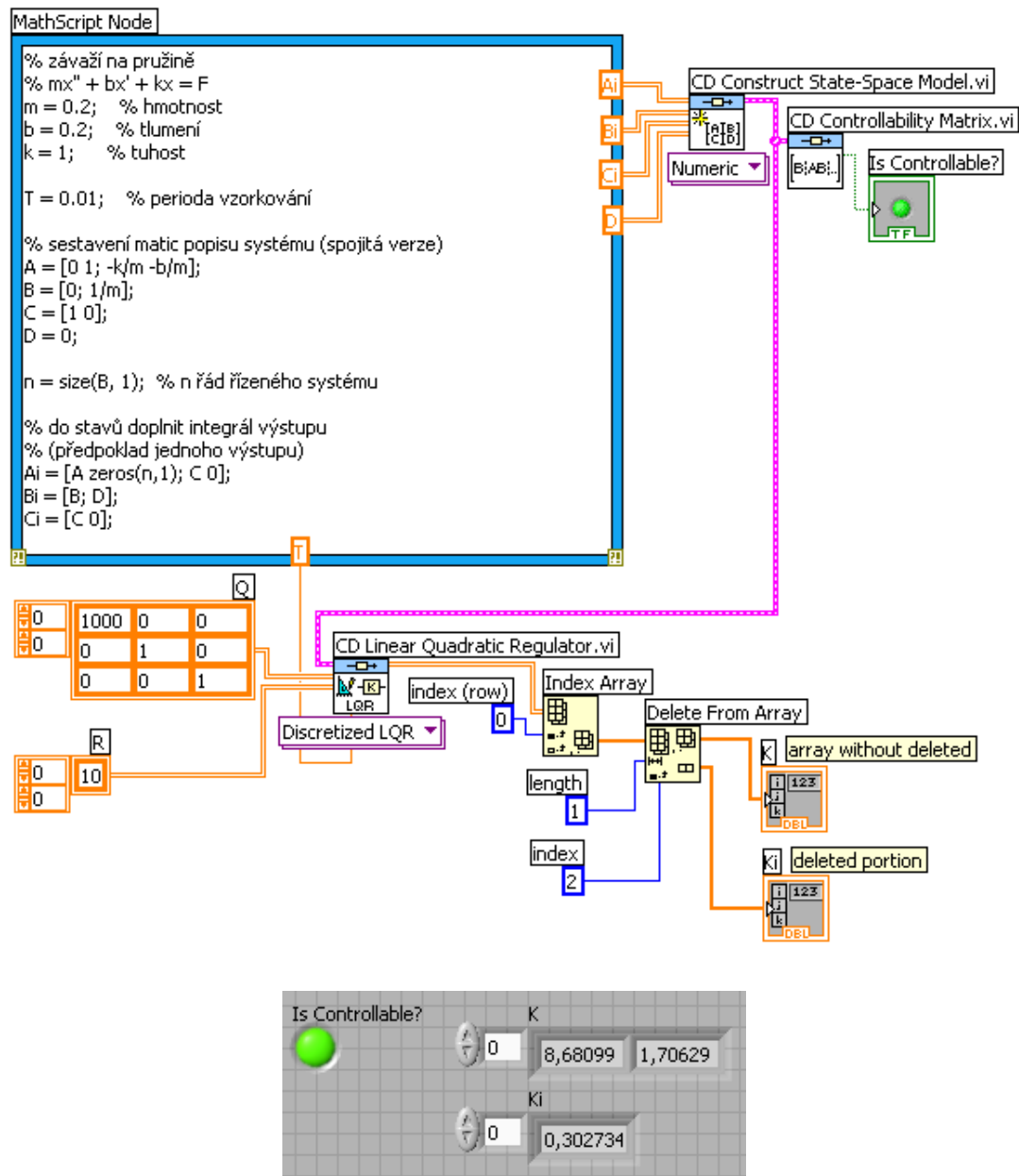
5 Stavové řízení s integrátorem na vstupu

Struktura řízení je stejná jako u LQ designu (viz. obrázek 2). Rozdíl mezi metodami spočívá ve způsobu návrhu zesílení regulátoru. V tomto případě se používá metoda volby pólů uzavřené smyčky. Metodu lze použít jak pro návrh zesílení spojitého regulátoru tak i diskrétního. Dále bude uvažován pouze diskrétní případ.

Akční zásah regulátoru je dán rovnicí $u[k] = -Kx[k]$. Dosazením do 2 a úpravou se získají rovnice popisující chování uzavřené smyčky.

$$\begin{aligned} x[k+1] &= (A - BK)x[k] \\ y[k] &= (C - DK)x[k] \end{aligned} \quad (8)$$

Obdobně jako při návrhu pozorovatele se zvolí požadované póly uzavřené smyčky a koeficienty vektoru zesílení \mathbb{K} se dopočítají tak, aby vlastní čísla (resp. póly uzavřené smyčky) matice $(A + \mathbb{B}\mathbb{K})$ byla rovna zvoleným pólům.



Obrázek 3: Příklad návrhu LQ regulátoru pomocí LabView

Při použití této metody lze velikost akčních zásahů ovlivnit pouze nepřímo. Platí, že čím více se volené póly uzavřené smyčky liší od pólů otevřené smyčky, tím větší jsou akční zásahy

navrženého regulátoru. Pokud není nutné podstatně změnit chování řízeného systému, ale je jej například potřeba pouze stabilizovat, je vhodné volit póly uzavřené smyčky tak, aby odezva uzavřené smyčky byla stabilní a o něco rychlejší než odezva otevřené smyčky. Tj. volené póly musí ležet uvnitř jednotkové kružnice (diskrétní model systému) a měly by ležet blíže k nule než póly otevřené smyčky. Navíc pokud nechceme, aby byla odezva uzavřené smyčky kmitavá, musí zvolené póly ležet na kladné polovině reálné osy.

Pro návrh zesílení regulátoru na základě zvolených pólů lze opět využít Matlab nebo LabView.

Tabulka 3: Příklad návrhu stavového regulátoru pomocí Matlabu

```
% závaží na pružině
%  $mx'' + bx' + kx = F$ 
m = 0.2; % hmotnost
b = 0.2; % tlumení
k = 1; % tuhost

T = 0.01; % perioda vzorkování

% sestavení matic popisu systému (spojitá verze)
A = [0 1; -k/m -b/m];
B = [0; 1/m];
C = [1 0];
D = 0;

n = size(B, 1); % n řád řízeného systému
m = size(B, 2); % m počet vstupních veličin

% do stavů doplnit integrál výstupu (předpoklad jednoho výstupu)
Ai = [A zeros(n,1); C 0];
Bi = [B; D];
Ci = [C 0];

sys = ss(Ai,Bi,Ci,D); % spojitý model

% kontrola říditelnosti
Mc = ctrb(sys);
ni = size(Bi, 1); % ni řád řízeného systému s integrálem
if (rank(Mc) ~= ni)
    error('System není říditelný.');
```

```

% póly otevřené smyčky jsou 1.0000, 0.9948 + 0.0217i a 0.9948 - 0.0217i
% póly uzavřené smyčky zvolit tak, aby odezva uzavřené smyčky byla
% stabilní, rychlejší a nekmitavá
cp = [0.95 0.9 0.85];

% spočítat odpovídající zesílení regulátoru
K = place(sys_d.A, sys_d.B, cp);

% z vektoru zesílení získat zesílení integrátoru
% (zesílení posledního stavu - integrálu polohy)
% (předpoklad jednoho výstupu)
Ki = K(ni) % Ki = 150.7575

% z vektoru zesílení odstranit zesílení integrátoru
K = K(1:n) % K = [52.7689 5.5537]

```

6 Stavové prediktivní řízení

Struktura řízení je opět stejná jako u předchozích dvou metod (viz. obrázek 2). Návrh zesílení regulátoru vychází z následující úvahy. Stav a výstup systému v k -tém časovém okamžiku jsou

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] \\ \mathbf{y}[k] &= \mathbf{C}\mathbf{x}[k] + \mathbf{D}\mathbf{u}[k], \end{aligned} \quad (9)$$

v k plus prvním časovém okamžiku jsou

$$\begin{aligned} \mathbf{x}[k+2] &= \mathbf{A}\mathbf{x}[k+1] + \mathbf{B}\mathbf{u}[k+1] = \mathbf{A}^2\mathbf{x}[k] + \mathbf{A}\mathbf{B}\mathbf{u}[k] + \mathbf{B}\mathbf{u}[k+1] \\ \mathbf{y}[k+1] &= \mathbf{C}\mathbf{x}[k+1] + \mathbf{D}\mathbf{u}[k+1] = \mathbf{C}\mathbf{A}\mathbf{x}[k] + \mathbf{C}\mathbf{B}\mathbf{u}[k] + \mathbf{D}\mathbf{u}[k+1], \end{aligned} \quad (10)$$

v k plus druhém časovém okamžiku jsou

$$\begin{aligned} \mathbf{x}[k+3] &= \mathbf{A}^3\mathbf{x}[k] + \mathbf{A}^2\mathbf{B}\mathbf{u}[k] + \mathbf{A}\mathbf{B}\mathbf{u}[k+1] + \mathbf{B}\mathbf{u}[k+2] \\ \mathbf{y}[k+2] &= \mathbf{C}\mathbf{A}^2\mathbf{x}[k] + \mathbf{C}\mathbf{A}\mathbf{B}\mathbf{u}[k] + \mathbf{C}\mathbf{B}\mathbf{u}[k+1] + \mathbf{D}\mathbf{u}[k+2] \end{aligned} \quad (11)$$

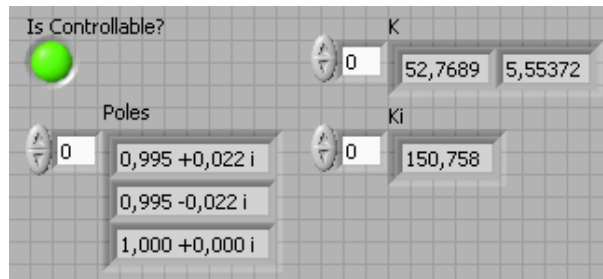
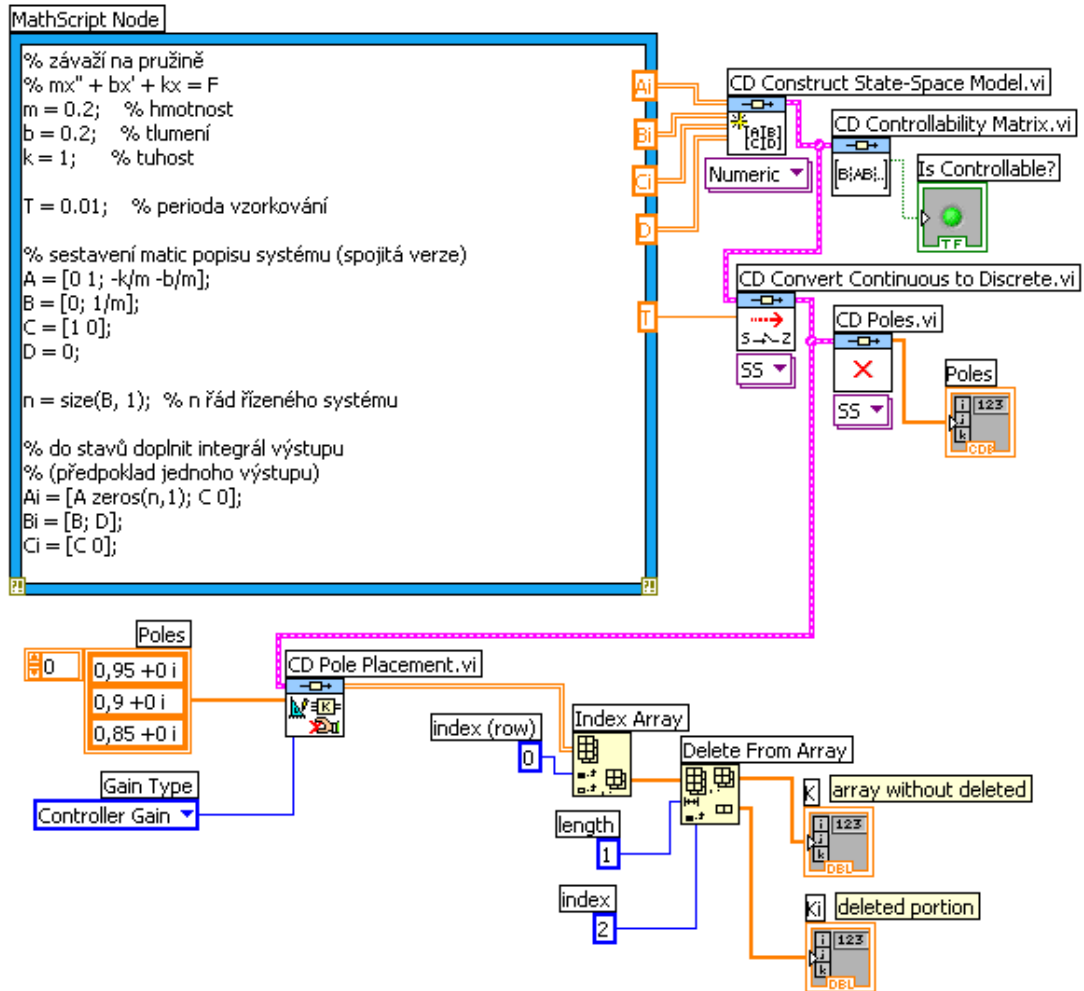
a tak dále. V $k+j$ -tém časovém okamžiku jsou tedy stavy a výstupy systému

$$\begin{aligned} \mathbf{x}[k+j+1] &= \mathbf{A}^{j+1}\mathbf{x}[k] + \sum_{i=0}^j \mathbf{A}^{j-1-i}\mathbf{B}\mathbf{u}[k+i] \\ \mathbf{y}[k+j] &= \mathbf{C}\mathbf{A}^j\mathbf{x}[k] + \sum_{i=0}^{j-1} \mathbf{C}\mathbf{A}^{j-1-i}\mathbf{B}\mathbf{u}[k+i] + \mathbf{D}\mathbf{u}[k+j]. \end{aligned} \quad (12)$$

Časový vývoj výstupů systému v závislosti na aktuálních stavech a budoucích vstupech může být zapsána jako

$$\mathbf{y} = \mathbf{F}\mathbf{x}[k] + \mathbf{H}\mathbf{u} \quad (13)$$

$$\text{kde } y = \begin{bmatrix} y[k] \\ y[k+1] \\ \vdots \\ y[k+j] \end{bmatrix}, u = \begin{bmatrix} u[k] \\ u[k+1] \\ \vdots \\ u[k+j] \end{bmatrix}, F = \begin{bmatrix} C \\ CA^1 \\ \vdots \\ CA^j \end{bmatrix}, H = \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{j-1}B & CA^{j-2}B & \dots & D \end{bmatrix}$$



Obrázek 4: Příklad návrhu stavového regulátoru pomocí LabView

Řídicí pravidlo neboli časová posloupnost vstupů systému \mathfrak{u} se získá minimalizací cenové funkce

$$J = (\mathfrak{y} - \mathfrak{w})^T \mathbb{Q} (\mathfrak{y} - \mathfrak{w}) + \mathfrak{u}^T \mathbb{R} \mathfrak{u}, \quad (14)$$

kde $\mathfrak{w} = \begin{bmatrix} \mathfrak{w}[k] \\ \mathfrak{w}[k+1] \\ \vdots \\ \mathfrak{w}[k+j] \end{bmatrix}$ je časový vývoj požadovaného výstupu systému a matice \mathbb{Q} a \mathbb{R} jsou

koeficienty cenové funkce. Dosazením do cenové funkce se získá

$$J = (\mathbb{F} \mathfrak{x}[k] + \mathbb{H} \mathfrak{u} - \mathfrak{w})^T \mathbb{Q} (\mathbb{F} \mathfrak{x}[k] + \mathbb{H} \mathfrak{u} - \mathfrak{w}) + \mathfrak{u}^T \mathbb{R} \mathfrak{u}. \quad (15)$$

Vstupy systému minimalizující cenovou funkcí jsou

$$\mathfrak{u} = \mathbb{L} (\mathfrak{w} - \mathbb{F} \mathfrak{x}[k]), \quad (16)$$

kde

$$\mathbb{L} = (\mathbb{H}^T \mathbb{Q} \mathbb{H} + \mathbb{R})^{-1} \mathbb{H}^T \mathbb{Q}. \quad (17)$$

Přestože tato metoda dává potřebné akční zásahy na několik kroků dopředu (v závislosti na počtu kroků predikce), v praxi je použitelný pouze první z nich, protože se na systému projevují vlivy poruch a nepřesností modelu. Následující akční zásah se musí spočítat znovu na základě aktuálních hodnot stavů. Za předpokladu, že požadovaný výstup je v čase konstantní, je pro výpočet aktuálního akčního zásahu třeba použít pouze tolik prvních řádků matice \mathbb{L} a součinu $(\mathbb{L}\mathbb{F})$ kolik je vstupů systému.

$$\mathfrak{u}[k] = \begin{bmatrix} u_1[k] \\ u_2[k] \\ \vdots \\ u_n[k] \end{bmatrix} = \begin{bmatrix} \sum_{t=1}^m (w_t[k] \sum_{i=1}^j L_{1,m(i-1)+t}) \\ \sum_{t=1}^m (w_t[k] \sum_{i=1}^j L_{2,m(i-1)+t}) \\ \vdots \\ \sum_{t=1}^m (w_t[k] \sum_{i=1}^j L_{n,m(i-1)+t}) \end{bmatrix} - \begin{bmatrix} \sum_{p=1}^s (LF)_{1,p} x_p[k] \\ \sum_{p=1}^s (LF)_{2,p} x_p[k] \\ \vdots \\ \sum_{p=1}^s (LF)_{n,p} x_p[k] \end{bmatrix} \quad (18)$$

kde n je počet vstupů systému, m je počet výstupů systému, j je počet kroků predikce, s je počet stavů systému, $u_n[k]$ je n -tý vstup systému, $w_t[k]$ je požadovaná hodnota t -tého výstupu systému, L označuje prvky matice \mathbb{L} , (LF) označuje prvky matice $(\mathbb{L}\mathbb{F})$ a $x_p[k]$ je p -tý stav systému.

Pro získání zesílení integrátoru je tentokrát třeba nejen přidat integrál výstupu mezi stavy, ale také mezi výstupy, jinak metoda navrhne nulové zesílení integrátoru.

Tato metoda návrhu stavového řízení není v žádném mě známém softwaru přímo podporována, ale protože jsou její podstatou výpočty s maticemi, je možné použít jakýkoli software, který umí provádět výpočty s maticemi, např. Matlab.

Tabulka 4: Příklad návrhu stavového prediktivního regulátoru pomocí Matlabu

```
% závaží na pružině
%  $mx'' + bx' + kx = F$ 
m = 0.2; % hmotnost
b = 0.2; % tlumení
k = 1; % tuhost

T = 0.01; % perioda vzorkování

% sestavení matic popisu systému (spojitá verze)
A = [0 1; -k/m -b/m];
B = [0; 1/m];
C = [1 0];
D = 0;

n = size(B, 1); % n řád řízeného systému
m = size(B, 2); % m počet vstupních veličin

% do stavů doplnit integrál výstupu (předpoklad jednoho výstupu)
Ai = [A zeros(n,1); C 0];
Bi = [B; D];
% doplnit integrál jako druhý výstup
Ci = [C 0; zeros(1,n) 1];
Di = [D; 0];

sys = ss(Ai,Bi,Ci,Di); % spojitý model

% kontrola říditelnosti
Mc = ctrb(sys);
ni = size(Bi, 1); % n řád řízeného systému s integrálem

if (rank(Mc) ~= ni)
    error('System není říditelný.');
```

```

% počet kroků predikce
j = 10;

% sestavit matici F z matematického odvození
for i=1:j
    cai = Cp*(Ap^(i-1)); % dvouřádková matice (výstup a integrál výstupu)
    F(2*i-1,:) = cai(1,:);
    F(2*i,:) = cai(2,:);
end;

% sestavit matici H z matematického odvození
H = zeros(2*j,j);
for i=1:j
    H(2*i-1,i) = Dp(1,1);
    H(2*i,i) = Dp(2,1);
end;

for sloupec=1:j
    for i=(sloupec+1):j
        % cab je dvouřádková matice (výstup a integrál výstupu)
        cab = Cp*(Ap^(i-(sloupec+1)))*Bp;
        H(2*i-1,sloupec) = cab(1,1);
        H(2*i,sloupec) = cab(2,1);
    end;
end;

% váhy cenové funkce
% velikost čtvercové matice Q je počet výstupů krát počet kroků predikce
Q = eye(2*j,2*j)*1e3;
% velikost čtvercové matice R je počet vstupů krát počet kroků predikce
R = eye(j,j);

% matice z matematického odvození
L = inv(H'*Q*H + R)*H'*Q;

% vybrat pouze (jeden, protože je pouze jeden vstup) řádek pro
% aktuální akční zásah (budoucí zanedbat, protože se
% budou počítat znovu)
pom = L*F;
K = pom(1,:);
Ki = K(3) % zesílení integrátoru
K = K(1:2) % stavová zesílení

% požadované hodnoty výstupů jsou 0 → není třeba počítat zesílení
% požadované hodnoty výstupu

```