

EXPERIMENTÁLNÍ IDENTIFIKACE STAVOVÉHO MODELU

F. Dušek, P. Škrabánek, J. Mareš

Univerzita Pardubice, Fakulta elektrotechniky a informatiky, Katedra řízení procesů,
Studentská 95, 532 10 Pardubice, Česká Republika
e-mail: frantisek.dusek@upce.cz

Abstrakt

Článek pojednává o využití prostředí MATLAB při získávání matematického popisu dynamického chování vícerozměrového reálného systému ve formě lineárního stavového modelu. Reálný systém představuje laboratorní zařízení Hydraulicko-pneumatická soustava (HPS). Prostředí MATLAB je využito nejen pro offline vyhodnocení naměřených dat, ale i pro provedení vlastního dynamického experimentu pro získání naměřených dat.

Abstract

The article deals with usage of MATLAB in derivation of real multidimensional system dynamical behavior description (in the form of linear state model). The real system is laboratory model of Hydraulic-pneumatic system (HPS). MATLAB is used not only for offline measured data evaluation but for the experiment realization too.

1 Úvod

Pro potřeby návrhu řízení je potřeba informace o vlastnostech a dynamickém chování sledovaného systému. Pod pojmem dynamické chování systému (zařízení) je myšlen popis časového průběhu výstupních veličin v závislosti na aktuálním stavu (počátečních podmínkách) a známém časovém průběhu vstupních veličin. Nejčastěji je tato závislost vyjádřena soustavou diferenciálních (nebo diferenčních) rovnic. Tyto rovnice se obvykle získají matematicko fyzikální analýzou (spojitý popis tj. diferenciální rovnice) nebo vyhodnocením experimentálně získaných dat – experimentální identifikací (diskrétní popis tj. diferenční rovnice). V případě experimentální identifikace jde o dvě úlohy. První je určení struktury (tvaru) rovnic – v případě lineárního popisu počet rovnic (řád systému). Druhou úlohou je odhad parametrů zvolené struktury. Dále v článku se budeme zabývat pouze experimentální identifikací diskrétního lineárního vícerozměrového systému popsaného stavovým modelem.

V kapitole „Reálné zařízení“ je stručně zmíněn popis soustavy, ze kterého vyplyne jak struktura modelu – systém s šesti stavovými veličinami (systém šestého řádu) se třemi vstupy a dvěma výstupy, tak i určité problémy s generováním vstupů a měřením výstupů. Dále je uveden stručný popis a řešení komunikace řídicí jednotky zařízení a programového prostředí MATLAB pod OS Windows.

V kapitole „Teoretická část“ je uveden konkrétní tvar matematického modelu pracující s veličinami ve fyzikálním významu a způsob výpočtu matic parametrů diskrétního stavového modelu na základě naměřených dynamických průběhů. Pro snížení nelinearity modelu je využito znalosti odmocninového charakteru závislosti průtoku ventilem na tlaku a tři měřené vstupní veličiny zařízení jsou přepočítány na dvě vstupní veličiny modelu.

V kapitole „Experimentální část“ je popsán dynamický experiment – generování tří průběhů žádaných hodnot tří regulačních obvodů (z nichž jsou dva realizovány v MATLABu)

a měření devíti veličin. Hodnoty měřených a akčních veličin jsou s intervalem jedna sekunda ukládány do textového souboru. Celý experiment probíhající v reálném čase je realizován v prostředí MATLABu.

Po dokončení experimentu navazuje vyhodnocení dat včetně optimalizace hodnot offsetů na měřených veličinách.

2 Reálný systém

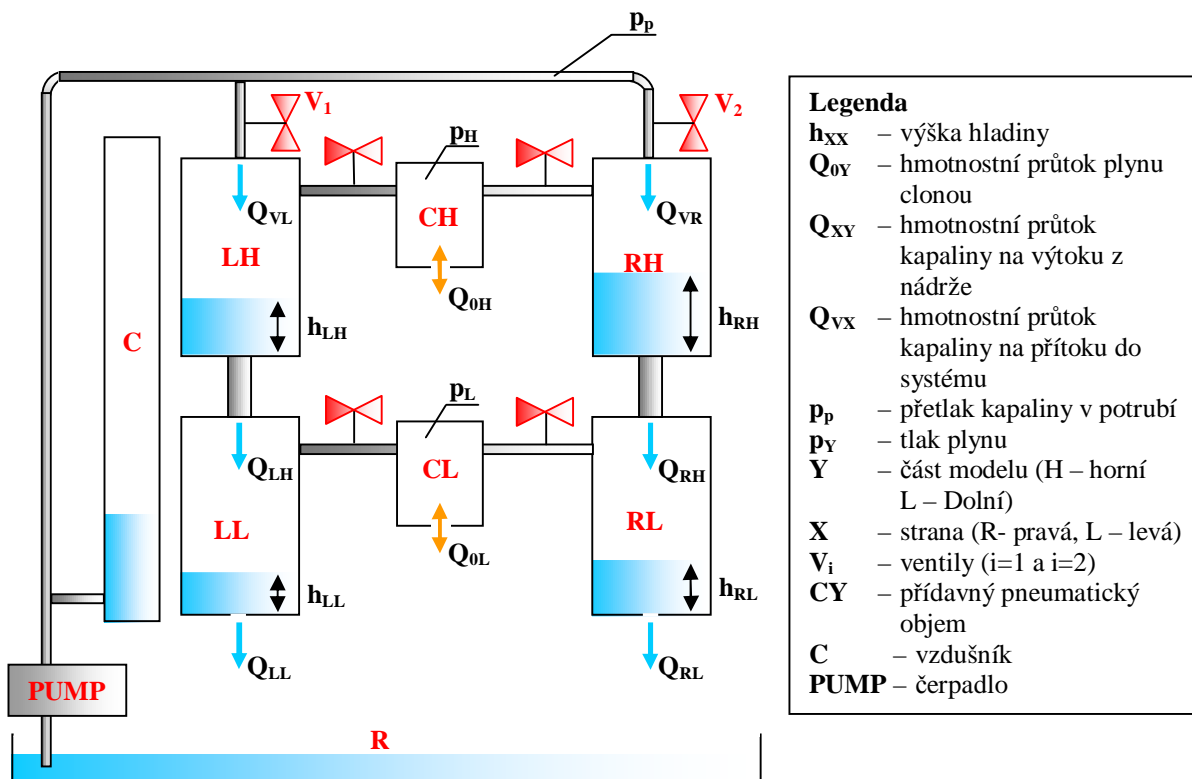
Dynamický popis se určoval pro laboratorní zařízení – Hydraulicko - pneumatickou soustavu (HPS), jejichž zjednodušené provedení bylo popsáno např. v [1], [2] a [3]. Použité zařízení se liší tím, že obsahuje pouze jedno čerpadlo se vzdušníkem a kapalina je vedena do horních nádrží přes ovládané ventily. Další podstatnou odlišností je, že jsou měřeny všechny hladiny a tlaky. Zařízení bylo částečně popsáno i v [4]. Kromě toho je vybaveno vlastní řídicí jednotkou (URJ), která kromě zajištění měření a ovládání může i regulovat tlak v přívodním potrubí. Tato jednotka také zajišťuje přenos dat (komunikaci) prostřednictvím sériové linky s vlastním komunikačním protokolem.

2.1 Popis zařízení

Zařízení se skládá celkem ze čtyř hydraulických nádrží (LH, RH, LL a RL) a dvou přídatných pneumatických objemů (CH a CL). Hydraulické nádrže v horní části modelu jsou navzájem spojeny přes společný pneumatický prostor CH a tudíž se vzájemně ovlivňují. Stejným způsob jsou propojeny i nádrže ve spodní části, což je dobře vidět na obrázku 1.

Systém má celkem tři ovladatelné vstupy. Prvním vstupem je výkon čerpadla. Další dva vstupy jsou otevření přítokových ventilů V_1 a V_2 . Výkon čerpadla lze ovládat dvěma způsoby. První je přímý způsob, kdy lze přímo zadávat požadovaný výkon. Druhý způsob je nepřímý. URJ je totiž vybavena PID regulátorem, který je schopen zajistit udržování požadované hodnoty tlaku kapaliny v přívodním potrubí. Dynamické vlastnosti regulace tlaku kapaliny jsou ovlivňovány vzdušníkem C. Soustava je opatřena pouze jedním čerpadlem a proto změna tlaku kapaliny v přívodním potrubí ovlivňuje zároveň přítok do pravé i levé části HPS (Q_{VR} a Q_{VL}).

Každá z hydraulických nádrží je na dně opatřena odtokovým otvorem. Kapalina z horních nádrží odtéká do nádrží spodních, a tvoří tak jejich vstup. Přídatné pneumatické objemy jsou opatřeny clonou s otvorem o velmi malém průměru. Ta slouží k vyrovnávání pneumatických tlaků při dynamických dějích uvnitř soustavy. Soustava je opatřena osmi tlakovými čidly. Čtyři slouží k měření výšky hladiny (h_{LH} , h_{RH} , h_{LL} a h_{RL}), dvě k měření tlakového rozdílu vzduchu v horní a dolní části modelu (p_H a p_L), jedno pro měření přetlaku kapaliny v přívodním potrubí p_p a jedno k měření atmosférického tlaku p_A .



Obrázek 1: Náčes hydraulicko-pneumatické soustavy

2.2 Měření a ovládání

Na zařízení jsou měřeny hladiny ve všech čtyřech nádržích nepřímo pomocí diferenčních tlakových čidel TMDG 338 Z3H (0-3 kPa / 0-10 V). Dále jsou měřeny přetlaky vůči okolí v obou pneumatických objemech tlakovými čidly TMVG 418 Z3H (-2 až +10 kPa / 0-10 V), tlak v přívodním potrubí před ventily tlakovým čidlem TMG 518 Z3H (0-100 kPa / 0-10 V) a absolutní atmosférický tlak čidlem TMAG 518 Z3H (80-120 kPa, 0-10 V). Kapalínu do přívodního potrubí se vzdušníkem dodává ponorné čerpadlo Elegend (max 8 l/min, max 80 kPa, 2 A / 12 V) ovládané spínáním napájecího napětí výkonovým tranzistorem ovládaným řídicím signálem PWM (120 Hz, rozlišení 1/255). Každý napouštěcí ventil je ovládan modelářským servem HITEC 300 s rozsahem $\pm 60^\circ$ řízeným šířkou pulsů (0.7 ms = -60° ; 1.5 ms = 0° ; 2.3 ms = $+60^\circ$) generovaných s frekvencí 50 Hz (maximální povolená frekvence). V důsledku konstrukce ovládacích ventilů rozsahu plně zavřený a plně otevřený ventil odpovídá pouze určité podmnožina plného rozsahu ovládání. Tento problém je řešen kalibrací ovládání ventilů – určení minimální (maximální) hodnoty ovládacího signálu, při které začíná kapalina protékat (již se dále nezvyšuje průtok).

Měření výstupních napětí čidel, generování ovládacích signálů, realizaci regulace tlaku v přívodním potrubí, vyhodnocování pulsů případného průtokoměru, obsluhu ovládacího panelu včetně realizace logiky nastavení a obousměrnou (full duplex) sériovou komunikaci zajišťuje univerzální řídicí jednotka (UŘJ) postavená okolo mikrořadiče Philips 80C552. Jde o 8-bitový mikrořadič doplněný o 32 kB externí paměti programu (EPROM) a 32 kB externí paměti RAM. Vestavěný A/D převodník je 10-bitový (proto jsou hodnoty měřených veličin celočíselné v rozsahu 0-1023). Obdobně i rozsah akčních veličin byl použit stejný. Generování pulsů pro ovládání obou serv je softwarové s daným rozlišením. Ovládání aktuálního příkonu čerpadla je sice hardwarové (PWM) s rozlišením 0-255, ale povel je akceptován v rozlišení 0-1023. UŘJ také zajišťuje bezpečnostní funkci spočívající v tom, že pokud měřené hladiny v některé z nádrží překročí nastavenou hodnotu, všechny akční

veličiny se nastaví do nulového stavu a dojde k tzv. blokaci. To znamená, že je potřeba provést odblokování (spec. zpráva nebo ruční reset UŘJ), aby bylo možné měnit hodnoty akčních veličin.

Full duplex komunikace probíhá prostřednictvím zpráv s kódově transparentním protokolem dovolujícím přenášet různě dlouhé bloky binárních dat. Velikost přenášených zpráv je z důvodu volby velikosti vyrovnávacích bufferů na straně UŘJ omezena na celkovou délku 256 byte. Zpráva obsahující měřená data, aktuální hodnoty akčních veličin a stav prvků na ovládacím panelu je vysílána každou sekundu. Zprávy o nastavení akčních veličin (jedné či všech tří najednou) je možné přijmout kdykoliv.

2.3 Komunikace s MATLABem

Na straně PC je komunikace s UŘJ realizovaná pomocí standardního portu RS232. Základní komunikace je realizována DLL knihovnou využívající standardní funkce API Win32 pro asynchronní komunikaci. Tato knihovna zajišťuje spuštění samostatného vlákna (thread) na vyšší prioritě při dokončení příjmu zprávy. Toto řešení zajišťuje, že nedojde k přetečení přijímacího bufferu v případě, že aplikace nevyčítá přijatá data (po zapnutí napájení UŘJ posílá data bez ohledu na to, zda je někdo chce či nikoliv). Obdobně využití asynchronní komunikace při odesílání zprávy urychluje odezvu tím, že odesílaná zpráva se přenesení do výstupního bufferu a řízení se hned vrátí aplikaci (nečeká se na relativně dlouho trvající odeslání všech dat) – vlastní vyslání dat z výstupního bufferu probíhá na úrovni přerušení. Dále je komunikace popsána v [6].

Tato základní knihovna je doplněna o interface respektující strukturu datových zpráv zařízení HPS a využívající funkce knihoven MATLABu (datové struktury a konverzní funkce).

<code>[flg , stav] = MathHPS ('Open' , port , rychlost)</code>	inicializace komunikace
<code>[flg , stav] = MathHPS ('Close')</code>	povinné ukončení
<code>[flg , stav] = MathHPS ('Stav')</code>	dotaz na stav modulu
<code>[flg , stav , mes] = MathHPS ('Read')</code>	dotaz na poslední přijatou zprávu
<code>[flg , stav] = MathHPS ('Write' , mes)</code>	odeslání zprávy

Z hlediska komunikace s prostředím MATLABu byla zvolena varianta předávající data ve formě struktury tj. s pojmenovanými položkami.

Měřené hodnoty vrácené funkcí `[flg , stav , mes] = MathHPS ('Read')` jsou v proměnné `mes`, která má položky (strukturu) uvedené v tabulce 1.

Tabulka 1: STRUKTURA PROMĚNNÉ MES

Položka	Rozsah	Význam
P : []	0-1023 = 0-100 kPa	tlak v přívodním potrubí
PA : []	0-1023 = 80-120 kPa	absolutní atmosférický tlak
PL : []	0-1023 = -7 +10 kPa	přetlak v dolním pneum. objemu
PH : []	0-1023 = -7 +10 kPa	přetlak v horním pneum. objemu
PLH : []	0-1023 = 0-30 kPa	dif. tlak v levé horní nádrži (hladina)
PRH : []	0-1023 = 0-30 kPa	dif. tlak v pravé horní nádrži (hladina)
PLL : []	0-1023 = 0-30 kPa	dif. tlak v levé dolní nádrži (hladina)
PRL : []	0-1023 = 0-30 kPa	dif. tlak v pravé dolní nádrži (hladina)
Pot1 : []	0-1023 = 0-10 V	poloha potenciometru č.1
Pot2 : []	0-1023 = 0-10 V	poloha potenciometru č.2

Pot3: []	0-1023 = 0-10 V	poloha potenciometru č.3
Pot4: []	0-1023 = 0-10 V	poloha potenciometru č.4
Prutok: []	0-65535	počet pulsů/s
Prep1: []	'Local'/'Remote'	poloha přepínače ovládání servo 1
Prep2: []	'Local'/'Remote'	poloha přepínače ovládání servo 2
Prep3: []	'Local'/'Remote'	poloha přepínače ovládání čerpadlo
Prep4: []	'Manual'/'Automat'	poloha přepínače stavu regulace
Blokace: []	0/1	stav blokace
Servo1: []	0-1023 = ±60°	aktuální hodnota nastavení serva č.1
Servo2: []	0-1023 = ±60°	aktuální hodnota nastavení serva č.2
Cerpadlo: []	0-1023 = 0-100%	aktuální hodnota příkonu čerpadla
ZadTlakP: []	0-1023 = 0-25 kPa	akt. žádaná hodnota tlaku v přívod. potrubí

Po zavolání funkce jsou hodnoty poslední přijaté zprávy vráceny okamžitě (nečeká se na nová data). Hodnota proměnné `flag` indikuje zda data již byla jednou vyčtena či nikoliv. Tato vlastnost dovoluje jednoduchou synchronizaci s přijatými daty viz funkce `Mer.m`.

```
function data=Mer
% precti data s cekáním
flag=1;
while flag~=0,
    [flag, stav, data]=MathPS('Read');
    pause(0.01);
end
return
```

3 Teoretická část

V teoretické části jsou nejprve ukázány výpočetní vztahy pro určení parametrů (matic **A**, **B**, **C** a **D**) lineárního diskrétního stavového modelu ze známých průběhů vstupních, stavových a výstupních veličin. Dále je uvedeno plnění pracovních matic pro konkrétní případ včetně přepočítání měřených dat na data použitá v pracovních maticích a možnosti dopočítání offsetu při přepočtech měřených hodnot tlakových diferencí na výšky hladin.

Identifikace oblíbené formy ve tvaru matice přenosů (1) pomocí identifikace jednotlivých (jeden výstup, několik vstupů) přenosů $b_{ij}(z^{-1})/a(z^{-1})$ není možná, protože při nezávislé identifikaci jednotlivých přenosů (byť ze stejných naměřených dat) není možné zajistit, aby se u všech přenosů identifikovaly stejné parametry polynomu jmenovatele $a(z^{-1})$.

$$\mathbf{y}(z^{-1}) = \begin{bmatrix} y_1(z^{-1}) \\ \mathbf{M} \\ y_{n_y}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \frac{b_{1,1}(z^{-1})}{a(z^{-1})} & \mathbf{K} & \frac{b_{1,n_u}(z^{-1})}{a(z^{-1})} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \frac{b_{n_y,1}(z^{-1})}{a(z^{-1})} & \mathbf{K} & \frac{b_{n_y,n_u}(z^{-1})}{a(z^{-1})} \end{bmatrix} \begin{bmatrix} u_1(z^{-1}) \\ \mathbf{M} \\ u_{n_u}(z^{-1}) \end{bmatrix} \quad (1)$$

Je potřeba upozornit, že v případě HPS jde o nelineární systém, takže lineární stavový model je vždy pouze aproximací chování skutečného zařízení. Část nelineárního chování se odstraní použitím průtoků ventily Q_x jako vstupních veličin modelu.

3.1 Matematický model

V dalším textu se předpokládá diskrétní stavový model ve standardním tvaru tj.

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

$$\text{kde } \mathbf{x}(k) = \begin{bmatrix} h_{LH} \\ h_{RH} \\ p_H \\ h_{LL} \\ h_{RL} \\ p_L \end{bmatrix} \quad \mathbf{u}(k) = \begin{bmatrix} Q_L = o_L \cdot \sqrt{p} \\ Q_R = o_R \cdot \sqrt{p} \end{bmatrix} \quad \mathbf{y}(k) = \begin{bmatrix} h_{LL} \\ h_{RL} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$\mathbf{x}(k)$ je sloupcový vektor hodnot stavových veličin (výšky hladin v metrech a tlaky v pneumatických objemech v kPa) v k -tém okamžiku vzorkování

$\mathbf{u}(k)$ je sloupcový vektor hodnot vstupních veličin (otevření ventilů v rozsahu 0-1 korigované na tlak v přívodním potrubí v kPa) v k -tém okamžiku vzorkování

Při přepočtu vstupních veličin zařízení (tlak v přívodním potrubí p , otevření pravého o_R a levého o_L ventilu) je využita znalost odmocninového charakteru této závislosti. Konkrétní (konstantní) hodnoty parametrů přepočtu jsou zahrnuty jako součást identifikovaných parametrů (prvky matice \mathbf{B}).

3.2 Offline identifikace parametrů stavového modelu

Základem off-line identifikace je řešení přeúčené soustavy lineárních rovnic vycházející z přepisu stavového modelu do maticovém zápisu. Stavový model je nutno přepsat do tvaru oddělujícího data od parametrů

$$\begin{aligned} \mathbf{x}^T(k+1) &= \mathbf{d}_k \times \Theta_x & \mathbf{y}^T(k) &= \mathbf{d}_k \times \Theta_y \\ \mathbf{d}_k &= \begin{bmatrix} \mathbf{x}^T(k) & \mathbf{u}^T(k) \end{bmatrix} & \mathbf{d}_{k+1} &= \begin{bmatrix} \mathbf{x}^T(k+1) & \mathbf{u}^T(k+1) \end{bmatrix} \\ \Theta_x &= \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix} & \Theta_y &= \begin{bmatrix} \mathbf{C}^T \\ \mathbf{D}^T \end{bmatrix} \end{aligned} \quad (3)$$

$$\mathbf{X}_{N-1} = \mathbf{D}_{N-1} \times \Theta_x \quad \mathbf{Y}_N = \mathbf{D}_N \times \Theta_y$$

kde datové matice jsou

$$\mathbf{X}_{N-1} = \begin{bmatrix} \mathbf{x}^T(2) \\ \mathbf{x}^T(3) \\ \mathbf{M} \\ \mathbf{x}^T(N-1) \\ \mathbf{x}^T(N) \end{bmatrix} \quad \mathbf{D}_{N-1} = \begin{bmatrix} \mathbf{x}^T(1) & \mathbf{u}^T(1) \\ \mathbf{x}^T(2) & \mathbf{u}^T(2) \\ \mathbf{M} & \mathbf{M} \\ \mathbf{x}^T(N-2) & \mathbf{u}^T(N-2) \\ \mathbf{x}^T(N-1) & \mathbf{u}^T(N-1) \end{bmatrix} \quad (4)$$

$$\mathbf{Y}_N = \begin{bmatrix} \mathbf{y}^T(1) \\ \mathbf{y}^T(2) \\ \mathbf{M} \\ \mathbf{y}^T(N-1) \\ \mathbf{y}^T(k+N) \end{bmatrix} \quad \mathbf{D}_N = \begin{bmatrix} \mathbf{x}^T(1) & \mathbf{u}^T(1) \\ \mathbf{x}^T(2) & \mathbf{u}^T(2) \\ \mathbf{M} & \mathbf{M} \\ \mathbf{x}^T(N-1) & \mathbf{u}^T(N-1) \\ \mathbf{x}^T(N) & \mathbf{u}^T(N) \end{bmatrix}$$

Vlastní řešení tj. nalezení matic \mathbf{A} , \mathbf{B} , \mathbf{C} a \mathbf{D} je pak dáno vztahem

$$\Theta_x = \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix} = (\mathbf{D}_{N-1}^T \times \mathbf{D}_{N-1})^{-1} \times \mathbf{D}_{N-1}^T \times \mathbf{X}_{N-1}$$

$$\Theta_y = \begin{bmatrix} \mathbf{C}^T \\ \mathbf{D}^T \end{bmatrix} = (\mathbf{D}_N^T \times \mathbf{D}_N)^{-1} \times \mathbf{D}_N^T \times \mathbf{Y}_N$$
(5)

3.3 Optimalizace offsetu měření hladin

Vzhledem k tomu, že model má být použit k řízení reálného zařízení s omezeními ve stavu (výšky hladin), je potřeba, aby model poskytoval hodnoty ve fyzikálních jednotkách. Tento přístup též dovoluje zahrnout známé informace do procesu identifikace modelu. Základním předpokladem je převod mezi měřenými hodnotami (obvykle jednotky převodníku) a příslušnými fyzikálními jednotkami včetně kalibrace na konkrétní podmínky. V našem případě jde o kalibrace měření výšek hladin a kalibraci převodu hodnoty otevření ventilů na skutečné otevření. Hodnoty offsetu kalibrace výšek hladin se v průběhu času (dny) mění (bublina v přívodu k diferenčnímu čidlu tlaku, zanášení vstupního otvoru čidla atd.). Proto bylo základní určení parametrů modelu doplněno o odhad korekce offsetu převodu výška hladiny - naměřená data. Protože nás primárně zajímá výška hladin, určuje se offset měření hladin tak, aby se preferovala shoda výstupu modelu s průběhem měřených hladin (čtyři veličiny) na úkor shody průběhu tlaků (dvě veličiny). Zároveň se do takto identifikovaných offsetů také zahrnuje chyba linearizace závislá na poloze pracovního bodu, kolem kterého jsou experimentální data na reálném zařízení získávána.

Vlastní optimalizační výpočet pak představuje vyhledání takových hodnot h_{xy0} , aby průměrná absolutní odchylka průběhů hladin vypočtených z modelu h^m a průběhů přepočtených hladin h^* byla minimální, tedy

$$\min_{\substack{h_{LH0}, h_{RH0} \\ h_{LL0}, h_{RL0}}} \frac{1}{N} \sum_{k=1}^N \left[|h_{LH}^*(k) - h_{LH}^m(k)| + |h_{RH}^*(k) - h_{RH}^m(k)| + |h_{LL}^*(k) - h_{LL}^m(k)| + |h_{RL}^*(k) - h_{RL}^m(k)| \right] \quad (6a)$$

korigované hladiny h^*

$$\begin{aligned} h_{LH}^*(k) &= h_{LH}(k) - h_{LH0} & h_{RH}^*(k) &= h_{RH}(k) - h_{RH0} \\ h_{LL}^*(k) &= h_{LL}(k) - h_{LL0} & h_{RL}^*(k) &= h_{RL}(k) - h_{RL0} \end{aligned} \quad (6b)$$

výpočet parametrů A,B

$$\mathbf{x}(k) = \begin{bmatrix} h_{LH}^*(k) \\ h_{RH}^*(k) \\ p_H(k) \\ h_{LL}^*(k) \\ h_{RL}^*(k) \\ p_L(k) \end{bmatrix} \quad \mathbf{u}(k) = \begin{bmatrix} o_L(k) \cdot \sqrt{p(k)} \\ o_R(k) \cdot \sqrt{p(k)} \end{bmatrix} \quad (6c)$$

výpočet výstupů modelu h^m

$$\Theta_x = \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix} = (\mathbf{D}_{N-1}^T \times \mathbf{D}_{N-1})^{-1} \times \mathbf{D}_{N-1}^T \times \mathbf{X}_{N-1}$$

$$\mathbf{x}^m(k+1) = \mathbf{A} \cdot \mathbf{x}^m(k) + \mathbf{B} \cdot \mathbf{u}(k) \quad \mathbf{x}^m(0) = \mathbf{x}(0) \quad (6d)$$

4 Experimentální část

Cílem experimentů bylo získat časové průběhy vybraných veličin (tlaku v přívodním potrubí, výšky hladin ve čtyřech nádržích, tlaků v obou pneumatických objemech, otevření obou ventilů) jako odezvu na časové průběhy vstupních veličin (žádaný tlak v přívodním potrubí a žádané hladiny v horních nádržích). Skutečné vstupy modelu jsou průtoky oběma ventily, které byly dopočítány z naměřených dat. Z důvodu udržení stavů v pracovní oblasti, která je popsána matematickým modelem (nesmí dojít k přetečení ani podtečení), nebyly skutečné vstupy zařízení (příkon čerpadla, otevření obou ventilů) vybudovány přímo, ale nepřímo změnou žádaných hodnot.

4.1 Dynamický experiment

Při dynamickém experimentu byly měněny žádané hodnoty tlaku v přívodním potrubí (regulace vestavěným regulátorem tlaku v UŘJ) a žádané hodnoty výšky hladin v horních nádržích (programový PI regulátor). Průběhy žádaných hodnot byly vhodně zvoleny tak, aby systém byl vybudován v celém pracovním rozsahu. Tento způsob vedení experimentu (určení žádaných průběhů a nikoliv vlastních vstupních veličin) omezuje možnost vytečení nebo přetečení v jeho průběhu. Navíc z pohledu časového průběhu je systém vybudován zejména přibližně ve frekvenční oblasti, která odpovídá budoucímu využití modelu tj. model by měl být nejpřesnější v oblasti frekvencí, které se budou vyskytovat při řízení.

Měření skutečného tlaku v přívodním potrubí dovoluje zaznamenat situaci, kdy výkon čerpadla je nedostatečný k dosažení požadovaného tlaku při velkém průtoku. Další výhodou je, že není nutné určovat závislost mezi příkonem čerpadla a tlakem v přívodním potrubí. Vzhledem k normovanému otevření ventilů do intervalu 0-1 je důležité správné nastavení tohoto normování (fyzické hranice otevření ventilu kdy začíná/končí změna průtoku s fyzickým otevřením) [5].

4.2 Vyhodnocení experimentálních dat

Vyhodnocovaná data – dvě měření z 24.4.2008 pro rozsahy ventilů a korekce na skutečnou výšku hladiny. Kritérium – průměrná absolutní odchylka všech výšek hladin

d.mhLH=0.127;	d.mPLH=300;	% hladina/měření v j.p.
d.mhRH=0.180;	d.mPRH=458;	% hladina/měření v j.p.
d.mhLL=0.124;	d.mPLL=292;	% hladina/měření v j.p.
d.mhRL=0.148;	d.mPRL=359;	% hladina/měření v j.p.
d.oLmin=250;	d.oLmax=860;	% rozsah ovládání levého ventilu v j.p.
d.oRmin=200;	d.oRmax=790;	% rozsah ovládání prvního ventilu v j.p.

2.sada (990 hodnot) d.oLmin=250; d.oLmax=860; % ovládání levého ventilu (j.p.)
 d.oRmin=200; d.oRmax=790; % ovládání pravého ventilu (j.p.)

Kritérium před optimalizací: 0.0181

Kritérium po optimalizaci: 0.0177

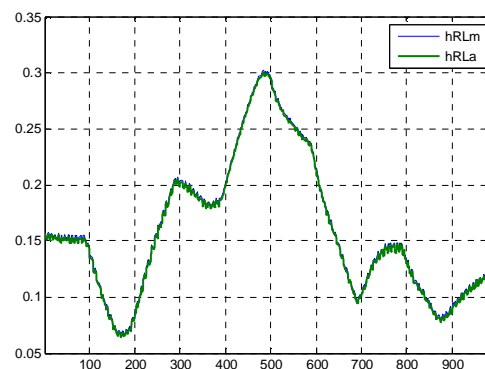
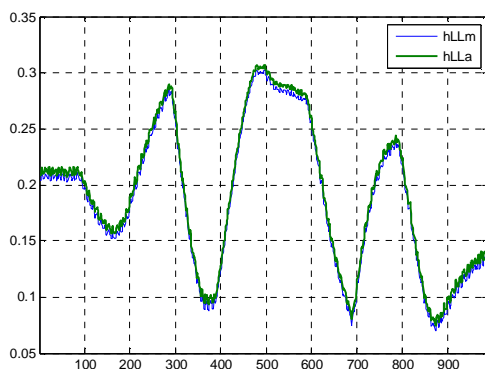
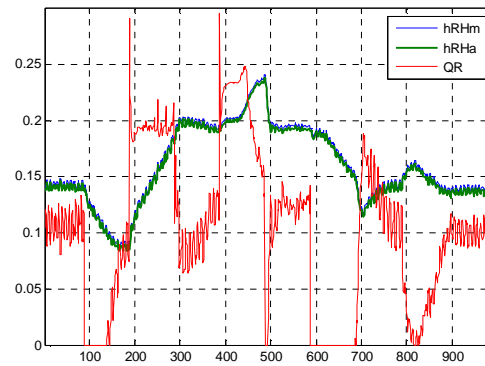
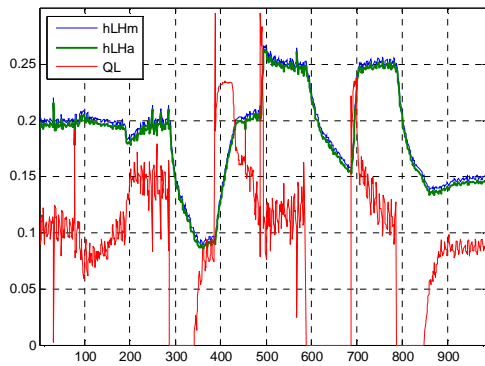
Posun: $h_{LH0}=0.006$ $h_{RH0}=0.004$ $h_{LL0}=-0.005$ $h_{LR0}=0.002$

A:

0.9902	-0.0053	0.0003	-0.0079	0.0047	-0.0025
0.0133	0.9831	0.0003	-0.0096	0.0074	-0.0012
-0.3077	-1.0914	1.0216	-1.3265	0.0169	-0.4637
0.0145	-0.0102	0.0003	0.9869	0.0076	0.0011
-0.0164	0.0227	0.0001	0.0172	0.9775	0.0015
0.0162	-0.2981	0.0125	-0.5302	0.0241	0.8458

B:

0.0022	-0.0004
-0.0010	0.0013
0.1542	0.0923
-0.0001	0.0001
0.0003	-0.0004
0.0432	0.0288



Obrázek 2: Grafické znázornění průběhu druhého experimentu

1.sada (810 hodnot) d.oLmin=276; d.oLmax=849; % ovládání levého ventilu (j.p.)
d.oRmin=214; d.oRmax=778; % ovládání pravého ventilu (j.p.)

Kritérium před optimalizací: 0.0155

Kritérium po optimalizaci: 0.0149

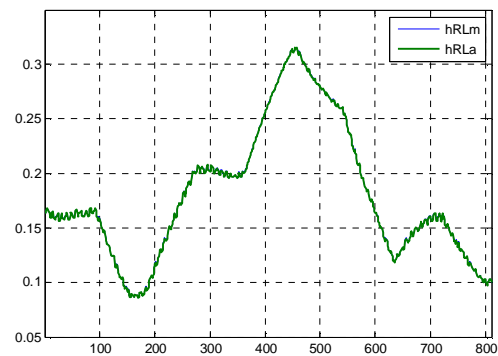
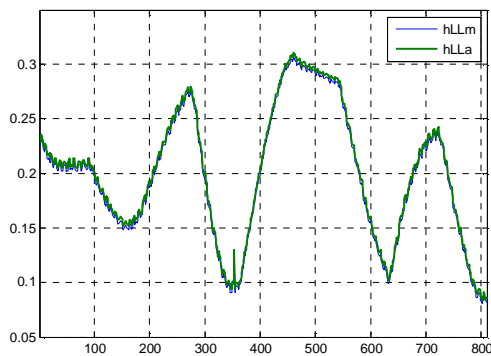
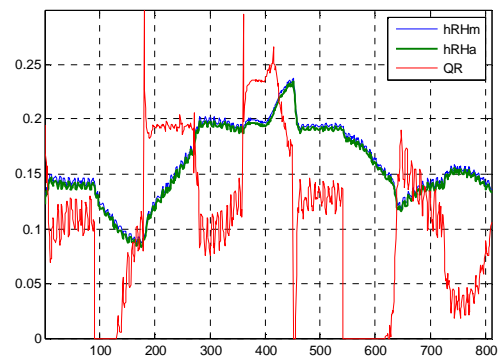
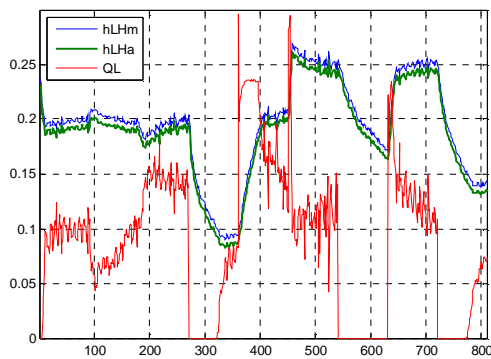
Posun: $h_{LH0}=0.007$ $h_{RH0}=0.007$ $h_{LL0}=-0.003$ $h_{LR0}=0.000$

A:

0.9891	-0.0052	0.0001	-0.0058	0.0060	-0.0017
0.0093	0.9792	0.0002	-0.0088	0.0083	-0.0015
-0.5072	-1.3538	0.9857	-1.3197	0.3004	-0.3932
0.0147	-0.0044	0.0005	0.9889	0.0035	0.0014
-0.0126	0.0212	0.0002	0.0151	0.9786	0.0009
-0.0736	-0.3227	0.0003	-0.5105	0.0615	0.8693

B:

0.0021	-0.0005
-0.0007	0.0014
0.1494	0.1122
-0.0000	-0.0003
0.0002	-0.0003
0.0447	0.0334



Obrázek 3: Grafické znázornění průběhu prvního experimentu

5 Závěr

Článek se zabývá problematikou získání matematického popisu dynamického chování reálného systému ve formě lineárního stavového modelu.

Nejprve je v článku popsáno maticové řešení offline odhadu parametrů lineárního vícerozměrového stavového modelu. Je popsána i modifikace řešení – optimalizace přepočtu měřených hodnot s cílem zvýšit přesnosti modelu ve smyslu lepší shody vybraných veličin s měřenými průběhy.

Dále je pozornost věnována vlastnímu experimentu, jehož specifikem je fakt, že součástí realizace jsou dva programové regulátory výšek hladin v horních nádržích soustavy, které musí zajišťovat nejen dostatečné vybuzení soustavy, ale zároveň i to, že v průběhu experimentu nedojde k úplnému vyprázdnění nebo přetečení některé ze čtyř nádrží. Poté následuje vyhodnocení a vlastní optimalizace dat pro získání stavového modelu.

Programové prostředí MATLAB bylo využito nejen pro offline vyhodnocení naměřených dat, ale i pro vlastní provedení dynamického experimentu pro získání naměřených dat. Jednoduchost řešení v MATLABu je dána koncepcí připojení využívající inteligentní řídicí jednotku zařízení, sériovou komunikaci a DLL knihovnu používající funkce API Windows pro asynchronní I/O operace.

Součástí článku jsou přílohy s výpisy programů doplněné o komentáře.

Práce byly provedeny s podporou výzkumného záměru MSM 0021627505 – část Řízení, optimalizace a diagnostika složitých systémů.

Literatura

- [1] MACHÁČEK, J.; DUŠEK, F.; HONC, D.: *Laboratory multivariable hydraulic-pneumatic process*. In.: International Carpathian Control Conference, 25.-28.5.2004, Zakopane 25-27, May 2004, Poland, Vol.I., p.133-138, ISBN 83-89772-00-0
- [2] HONC, D. - DUŠEK, F.: *Hydraulic-pneumatic system - technical data*. In: 6th International Scientific-Technical Conference Process Control 2004, Kouty nad Desnou 8-11, June 2004, Czech Republic, University of Pardubice, 2004, p. 265, ISBN 80-7194-662-1 (plný text 6 stran na doprovodném CD)
- [3] MACHÁČEK, J.; HONC, D.; DUŠEK, F. : *Výukový laboratorní model hydraulicko-pneumatické soustavy*. Automa 8-9, 2005, s. 108-109
- [4] ŠKRABÁNEK, P. : *Matematický model hydraulicko-pneumatické soustavy* [Diplomová práce]. Pardubice : Univerzita Pardubice, 2004. 51
- [5] ŠKRABÁNEK, P.; DUŠEK, F.: *Automatická identifikace parametrů ventilů*
In.: 14th Annual Conference Proceedings of Technical Computing Prague 2006, Kongresové centrum ČVUT, Praha, 26. Oct 2006, p.94, ISBN 80-7080-616-6 (plný text 6 stran na doprovodném CD)
- [6] DUŠEK, F.: *Jedno řešení sériové komunikace*. (V tomto sborníku).

Doc. Ing. František Dušek, CSc.

Univerzita Pardubice,
Fakulta elektrotechniky a informatiky, Katedra řízení procesů,
Studentská 95, 532 10 Pardubice, Česká Republika
e-mail: frantisek.dusek@upce.cz

Ing. Pavel Škrabánek

Univerzita Pardubice,
Fakulta elektrotechniky a informatiky, Katedra řízení procesů,
Studentská 95, 532 10 Pardubice, Česká Republika
e-mail: pavel.skrabanek@upce.cz

Ing. Jan Mareš

Univerzita Pardubice,
Fakulta elektrotechniky a informatiky, Katedra řízení procesů,
Studentská 95, 532 10 Pardubice, Česká Republika
e-mail: jan.mares2@student.upce.cz

6 Přílohy

6.1 Použití funkcí knihovny MatHPS.dll

seznam funkcí pro měření a ovládání HPS a význam jejich parametrů – obsah souboru MatHPS.m

```
% MatHPS.dll modul pro spojení modelu hydraulicko-pneumaticke
% soustavy (HPS) s MATLABem prostřednictvím seriové linky RS232.
% Format zprav je pevne dan a pro vlastní přenos je použit kódové
% transparentní protokol (UCTP).
% Použití modulu z MATLABu:
% *****      povinna inicializace komunikace
% [flg,stav]=MathHPS('Open',port,rychlost)
%   flg..... = 0 (inicializace O.K.), <>0 (chyba)
%   stav..... = (struktura) stavova slova dll_UCTP (UINT32)
%   port..... nazev portu ('COM1')
%   rychlost. prenosova rychlost (150,300,...,115200)
% *****      povinne ukonceni
% [flg,stav]=MathHPS('Close')
%   flg..... = 0 (uzavreni O.K.), <>0 (komunik. nebyla inic.)
%   stav..... = (struktura) stavova slova dll_UCTP (UINT32)
% *****      dotaz na stavova slova modulu
% [flg,stav]=MathHPS('Stav')
%   flg..... = 0
%   stav..... = (struktura) stavova slova dll_UCTP (UINT32)
% *****      dotaz na posledni prijatou zprvu
% [flg,stav,mes]=MathHPS('Read')
%   flg..... = 0 (zadna zprava) <> 0 (pocet dat)
%   stav..... = (struktura) stavova slova dll_UCTP (UINT32)
%   mes..... = (struktura) prijate zpravy
% *****      odeslani zpravy
% [flg,stav]=MathHPS('Write',mes)
%   flg..... = 0 (zapis O.K.) <>0 (chyba)
%   stav..... = (struktura) stavova slova dll_UCTP (UINT32)
%   mes..... vektor odeslane zpravy
%               =[1,0-1023] ... nastav servo 1
%               =[2,0-1023] ... nastav servo 2
%               =[3,0-1023] ... nastav cerpadlo/zadanou tlaku
%               =[4,0-1023,0-1023,0-1023] ... nastav vse
%               =[5] ... odblokuj
```

6.2 Pomocné funkce pro experiment

Definice parametrů

```
function d=defHPSpar
```

```
% Datum: 23.10.2005
```

```
% PARAMETRY PŘEPOČTU OVLÁDÁNÍ (SCHV 9.2.05)(ZadHPS.m + PrepMer.m) *****
d.oLmin=250;      d.oLmax=860;      % ovládání levého ventilu (j.p.)
d.oRmin=200;      d.oRmax=790;      % ovládání pravého ventilu (j.p.)
d.oCmin=86;      % ovládání čerpadla
% PARAMETRY KALIBRACE MĚŘENÍ HLADIN (12.10.05) (PrepMer.m) *****
d.mhLH=0.127;      d.mPLH=300;      % hladina/měření v j.p.
d.mhRH=0.180;      d.mPRH=458;      % hladina/měření v j.p.
d.mhLL=0.124;      d.mPLL=292;      % hladina/měření v j.p.
d.mhRL=0.148;      d.mPRL=359;      % hladina/měření v j.p.
% PARAMETRY KALIBRACE MĚŘENÍ TLAKŮ (PrepMer.m) *****
d.oPP=0;  d.oPH=3;  d.oPL=5;      % ofsety tlaků (j.p.)
```

Funkce pro přepočítání měřených hodnot v jednotkách převodníku na fyzikální jednotky

```
function dou=PrepMer(din,d)
% prep. mer. dat do inženýrských jednotek s parametry dle d (defHPSpar.m)
% 7.10.2005
% dou=PrepMer(din,d)

% ***** přepočítání z jednotek převodníku
% *** při použití MathPS.dll (převodník 0-1023)
mh = 0.3/1023/0.998/0.981; % přepočítání 0-1023 = 0-3 kPa na m H2O

% ***** vlastní přepočítání
dou=din;
dou.P = (din.P-d.oPP)/1023*100; % pP (0~100 kPa)
dou.PA = din.PA/1023*40+80; % pA (80~120 kPa)
dou.PL = (din.PL-d.oPL)/1023*12-2; % pL (-2~10 kPa)
dou.PH = (din.PH-d.oPH)/1023*12-2; % pH (-2~10 kPa)
dou.PLH = (din.PLH-d.mPLH)*mh+d.mhLH; % hLH (0~0.3 m)
dou.PRH = (din.PRH-d.mPRH)*mh+d.mhRH; % hRH (0~0.3 m)
dou.PLL = (din.PLL-d.mPLL)*mh+d.mhLL; % hLH (0~0.3 m)
dou.PRL = (din.PRL-d.mPRL)*mh+d.mhRL; % hRH (0~0.3 m)
dou.Servo1 = (din.Servo1-d.oLmin)/(d.oLmax-d.oLmin); % oL (0-1)
dou.Servo2 = (din.Servo2-d.oRmin)/(d.oRmax-d.oRmin); % oR (0-1)
dou.Cerpadlo=(din.Cerpadlo-d.oCmin)/(1023-d.oCmin); % xC (0-1)
```

Funkce pro nastavení hodnot na zařízení

```
function ZadHPS(typ,data,d)
% zápis dat do HPS, data v inženýrských jednotkách (procentech)
%
% ZadHPS(typ,data,d)
%

switch typ
case 1 % nastav servo 1
x=d.oLmin+(d.oLmax-d.oLmin)*data(1);
[flg,stav]=MathPS('Write',[typ,x]);
case 2 % nastav servo 2
x=d.oRmin+(d.oRmax-d.oRmin)*data(1);
[flg,stav]=MathPS('Write',[typ,x]);
case 3 % nastav cerpadlo
x=d.oCmin+(1023-d.oCmin)*data(1);
[flg,stav]=MathPS('Write',[typ,x]);
case 4 % nastav vse s cerpadlem
x1=d.oLmin+(d.oLmax-d.oLmin)*data(1);
x2=d.oRmin+(d.oRmax-d.oRmin)*data(2);
x3=d.oCmin+(1023-d.oCmin)*data(3);
[flg,stav]=MathPS('Write',[typ,x1,x2,x3]);
case 5 % odblokuj
[flg,stav]=MathPS('Write',typ);
case 6 % nastav tlak
x3=1023*data(3)/100;
[flg,stav]=MathPS('Write',[3,x3]);
case 7 % nastav vse s tlakem
x1=d.oLmin+(d.oLmax-d.oLmin)*data(1);
x2=d.oRmin+(d.oRmax-d.oRmin)*data(2);
x3=1023*data(3)/100;
[flg,stav]=MathPS('Write',[4,x1,x2,x3]);
otherwise
disp('Chybné číslo zpravy')
end
```

6.3 Skript pro dynamický experiment

Popis skriptu

- 1) inicializace a kontrola komunikace s HPS
- 2) kontrola nastavení ovládacího panelu, zda je možné vzdálené ovládání a je zapnuta regulace tlaku v přívodním potrubí
- 3) pomocí regulace hladin v horních nádržích dosáhnout počáteční hodnoty zadaného průběhu žádaného stavu a počkat do ustálení
- 4) provést regulaci na zadaný průběh žádaných hodnot hladin v horních nádržích a tlaku v přívodním potrubí, průběžně zaznamenávat do souboru potřebné hodnoty
- 5) ukončit komunikaci a vykreslit časový průběh vybraných veličin

```
clear
% skript pro měření (identifikace) HPS (F.D. 2.3.2008)
% měření vyžaduje MathHPS.dll + Mer.m + PrepMer.m + defHPSpar.m
% vyhodnocení vyžaduje IdentSSof.m + Control Tbx

% inicializace hardware *****
[flg,stav]=MathHPS('Open','COM6',9600);
if (flg==0) || (flg==2), disp('Komunikace s HPS navázána');
else disp('Neni komunikace s HPS'), return
end

data=Mer; % první měření
% kontrola stavu přepínačů *****
fprintf('\n***** Stav přepínačů připojovacího panelu *****\n');
s=0; sys=0; syd=0;
fprintf('Servo1 (ventil levý) poloha: %s\n',data.Prep1);
if data.Prep1(1)=='L',fprintf('!!! Přepni přepínač 1 do polohy Remote\n');
s=1; end
fprintf('Servo2 (ventil pravý) poloha: %s\n',data.Prep2);
if data.Prep2(1)=='L',fprintf('!!! Přepni přepínač 2 do polohy Remote\n');
s=1; end
fprintf('Čerpadlo (ovládání) poloha: %s\n',data.Prep3);
if data.Prep3(1)=='L',fprintf('!!! Přepni přepínač 3 do polohy Remote\n');
s=1; end
fprintf('Čerpadlo (režim) poloha: %s\n',data.Prep4);
if data.Prep4(1)=='M',fprintf('!!! Přepni přepínač 4 do polohy Automat\n');
s=1; end
fprintf('Blokace HPS stav: %s\n',data.Blokace);
if s==1, [flg,stav]=MathHPS('Close'); return; end
[flg,stav]=MathHPS('Write',[5]); % pro jistotu odblokuj

% priprava prubehu zadane *****
x=ones(1,100); % základní interval
W=[x*0.20;x*0.15;x*25]; % definuje výchozí ustálený stav
W=[W,[x*0.2;x*0.1;x*35]]; W=[W,[x*0.2;x*0.25;x*15]];
W=[W,[x*0.1;x*0.20;x*35]]; W=[W,[x*0.20;x*0.25;x*30]];
W=[W,[x*0.25;x*0.20;x*40]]; W=[W,[x*0.15;x*0.15;x*20]];
W=[W,[x*0.25;x*0.15;x*25]]; W=[W,[x*0.15;x*0.15;x*15]];
W=[W,[x*0.15;x*0.15;x*15]]; [x,N]=size(W);

par=defHPSpar; % zadané parametry HPS
% nastavení ustáleného stavu *****
% na první hodnoty z průběhu žádané se regulují horní hladiny a čeká se
% na ustálení dolních hladin (změna výstupu exponenciálního filtru)
iL=0; iR=0; konec=1; X0=zeros(6,1); U0=zeros(3,1);
```

```

t=1:60; XX=zeros(6,60); % 1 min historie pro zobrazení
while konec>0,
    data=Mer; data=PrepMer(data,par); % měření
    X=[data.PLH;data.PRH;data.PH;data.PLL;data.PRL;data.PL];
    % regulátor hladina hLH
    e=W(1,1)-X(1); iL=iL+e;
    if iL>1, iL=1; end, if iL<0, iL=0; end
    oL=e*20+iL*0.75; % PI-reg
    if oL<0, oL=0; end if oL>1, oL=1; end
    % regulátor hladina hRH
    e=W(2,1)-X(2); iR=iR+e;
    if iR>1, iR=1; end if iR<0, iR=0; end
    oR=e*20+iR*0.35; % PI-reg
    if oR<0, oR=0; end if oR>1, oR=1; end
    U=[oL;oR;W(3,1)]; % vektor aktuálních akčních
    ZadHPS(7,U,par); % [0-1,0-1,kPa] % nastav akční
    X0=0.9*X0+0.1*X; U0=0.9*U0+0.1*U; % filtrace měření
    XX=[XX(:,2:end),X]; % aktualizace a posun historie pro zobrazení
    plot(t,XX(1,:),t,XX(2,:),t,XX(4,:),t,XX(5,:))
    axis([0,60,0.01,0.3]), legend('hLH','hRH','hLL','hRL')
    if abs(X0(4:5,1)-XX(4:5,1))<0.003, konec=0; end % ustálení dole ?
end
fprintf('*****\n');
fprintf('oL=%5.3f oR=%5.3f p=%5.3f\n',U0);
disp('Ustálený stav - Start mereni Enter'); pause

% vlastní experiment *****
XX=zeros(N,7); UU=zeros(N,3); XK=zeros(7,60);
%iL=U0(1)/0.2; iR=U0(2)/0.2;
ff=fopen('datHPS.txt','w'); % otevři soubor pro zápis
fprintf(ff,'***** Data HPS mereni: %s *****\r\n',datestr(now));
fprintf(ff,'*hLH** *hRH** **pH** *hLL** *hRL** **pL** **P*** **oL** **oR**
**PW**\r\n');
for k=1:N,
    data=Mer; data=PrepMer(data,par); % měření
    X=[data.PLH;data.PRH;data.PH;data.PLL;data.PRL;data.PL;data.P];
    % regulátor hladina hLH
    e=W(1,k)-X(1); iL=iL+e;
    if iL>1, iL=1; end if iL<0, iL=0; end
    oL=e*20+iL*0.75; % PI-reg
    if oL<0, oL=0; end if oL>1, oL=1; end
    % regulátor hladina hRH
    e=W(2,k)-X(2); iR=iR+e;
    if iR>1, iR=1; end if iR<0, iR=0; end
    oR=e*20+iR*0.3; % PI-reg
    if oR<0, oR=0; end if oR>1, oR=1; end
    U=[oL;oR;W(3,k)];
    ZadHPS(7,U,par); % [0-1,0-1,kPa]
    XK=[XK(:,2:end),X]; % aktualizace a posun historie pro zobrazení
    plot(t,XK(1,:),t,XK(2,:),t,XK(4,:),t,XK(5,:))
    axis([0,60,0.05,0.3]), legend('hLH','hRH','hLL','hRL')
    pri=[W(1,k),W(2,k),N-k,W(3,k)];
    fprintf(ff,'%6.3f %6.3f %6.2f %6.3f %6.3f %6.2f %6.2f %6.3f %6.3f
%6.2f\r\n',[X;U]);
    XX(k,:)=X'; UU(k,:)=U';
end
fclose(ff);
ZadHPS(7,[0,0,0],par); % vypni [0-1,0-1,kPa]
pause(2)
[flg,stav]=MathPS('Close');

```



```

% vykreslení experimentu *****
t=1:N;
subplot(221), plot(t,XX(:,1:2),t,XX(:,4:5)), title('hladiny')
subplot(222), plot(t,XX(:,3),t,XX(:,6)), title('tlaky pH,pL')
subplot(223), plot(t,UU(:,1:2)), title('otevreni')
subplot(224), plot(t,UU(:,3)), title('tlak P')

```

6.4 Vyhodnocení dat – identifikace a optimalizace

```

% skript na určení stavového modelu HPS z datHPS.txt
% x(k+1)=A*x(k)+B*u(k)
% *hLH** *hRH** **pH** *hLL** *hRL** **pL** **P** **oL** **oR** **PW**

```

```

load datHPS.txt
d=datHPS;
clear datHPS
[N,x]=size(d); t=1:N;
Xd=d(:,1:6);
P=sqrt(d(:,7)); oL=d(:,8); oR=d(:,9);
Ud=[oL.*P,oR.*P];
D=[Xd(1:end-1,:),Ud(1:end-1,:)]; X=Xd(2:end,:);
V=inv(D'*D)*D'*X;
A=V(1:6,:); B=V(7:8,:);
p=[0;0;0;0];
[Kr,Xa]=fVyhData(p,Xd,Ud);
fprintf(' Kriterium = %8.4f\n',Kr);
fprintf('hLH0=%5.3f hRH0=%5.3f hLL0=%5.3f hLR0=%5.3f \n',p)
p=fminsearch(@(p) fVyhData(p,Xd,Ud),p);
[Kr,Xa]=fVyhData(p,Xd,Ud);
fprintf(' Kriterium = %8.4f\n',Kr);
fprintf('hLH0=%5.3f hRH0=%5.3f hLL0=%5.3f hLR0=%5.3f \n',p)
X0=[p(1:2);0;p(3:4);0];
Xv=Xd-ones(N,1)*X0;
subplot(221), plot(t,Xd(:,1),t,Xv(:,1),t,Ud(:,1)/20)
legend('hLHm','hLHa','QL'), grid, axis([1,N,0.0,0.3])
subplot(222), plot(t,Xd(:,2),t,Xv(:,2),t,Ud(:,2)/20)
legend('hRHm','hRHa','QR'), grid,axis([1,N,0.0,0.3])
subplot(223), plot(t,Xd(:,4),t,Xv(:,4))
legend('hLLm','hLLa'), grid, axis([1,N,0.05,0.35])
subplot(224), plot(t,Xd(:,5),t,Xv(:,5))
legend('hRLm','hRLa'), grid, axis([1,N,0.05,0.35])

D=[Xv(1:end-1,:),Ud(1:end-1,:)]; X=Xv(2:end,:);
V=inv(D'*D)*D'*X;
A=V(1:6,:); B=V(7:8,:);

```

Výpočet kritéria optimalizace – predikce z modelu a rozdíl od naměřených průběhů

```

function [Kr,Xa]=fVyhData(p,Xd,Ud)
%
Xm=Xd; Xa=Xd;
[N,x]=size(Xd);
X0=[p(1:2);0;p(3:4);0];
Xd=Xd-ones(N,1)*X0;
D=[Xd(1:end-1,:),Ud(1:end-1,:)]; X=Xd(2:end,:);
V=inv(D'*D)*D'*X;
A=V(1:6,:); B=V(7:8,:);
X=Xd(1,:);
for k=1:N-1,
    X=A*X+B*Ud(k,:);

```

```
    Xa(:,k+1)=X;  
end  
X=sum(abs(Xm-Xa),2);  
Xa=Xa';  
Kr=(X(1)+X(2)+X(4)+X(5))/N/4;
```