

# IDENTIFIKACE A LINEARIZACE V SIMULINKU

*František Dušek, Daniel Honc*

Katedra řízení procesů a výpočetní techniky, FCHT, Univerzita Pardubice

## 1. Úvod

Vhodné programové vybavení může výrazně usnadnit zpracování experimentálních dat i odstranit pracné analytické odvozování. V článku je ukázáno využití MATLABu v kontextu řešení konkrétního případu – návrhu řízení teploty pece. Jde o problém určení parametrů matematického nelineárního dynamického modelu reálné soustavy z experimentálně změřených dat a získání sady linearizovaných stavových modelů pro účely návrhu řízení. V obou úlohách je použit stejný nelineární dynamický model soustavy realizovaný jako model v SIMULINKu. První úloha představuje použití výpočetní síly pro zpracování většího množství dat – vyhledání takových hodnot 13 parametrů soustavy 4 diferenciálních rovnic, které minimalizují odchylku experimentálně změřeného průběhu výstupu soustavy od výstupu modelu. Úloha druhá pak ukazuje řešení problému nalezení stavu modelu odpovídajícího zadaným podmínkám (řešení nelineárních algebraických rovnic) a linearizaci soustavy nelineárních diferenciálních rovnic (vyčíslení matice parciálních derivací – jacobianu pro určený stav). Popis celého řešení je uveden pouze pro názornost a části nesouvisející s použitím MATLABu jsou zkráceny.

## 2. Cíl práce a popis řešení

Úkolem bylo navrhnout řízení teploty chemického reaktoru umístěného v elektricky vyhřívané peci. Teplota reaktoru je měřena a cílem řízení je, aby teplota reaktoru sledovala dopředu daný časový průběh. Akční veličinou je příkon topné spirály pece. Problémem v tomto případě bylo nezanedbatelné nelineární chování v požadovaném pracovním rozsahu teplot (cca 20 – 1000 °C). Díky nelinearitě soustavy není možné dosáhnout při standardní regulaci s PID regulátorem požadované kvality sledování žádaného průběhu teploty v celém rozsahu. Lepší kvality sledování je možné s použitím složitějších regulačních algoritmů, které ale vyžadují znalost matematického modelu řízené soustavy. Matematický model celý problém komplikuje nejen tím, že je nutné určit jeho strukturu (např. matematickofyzikální analýzou) ale i tím, že je nutné určit hodnoty všech jeho parametrů. Některé hodnoty sice mohou vyplýnout z fyzikálních vlastností či konstrukčního provedení, ale většinou je potřeba jejich hodnoty určit ze znalosti naměřených dat na reálném zařízení (experimentální identifikace).

Další komplikací návrhu řízení nelineárních soustav je, že standardní metody návrhu řízení vycházejí z lineárního modelu. Proto se obvykle provádí linearizace nelineárního modelu ve vybraném pracovním bodě. Záleží pak na konkrétním případě, zda zhoršení kvality regulace vyvolané chybou linearizace je či není akceptovatelné.

V případě řízení teploty pece byla pro částečné potlačení nelinearity využita jedna z metod adaptivního řízení – Gain Scheduling (GS). Princip GS spočívá v tom, že se pro vybrané stavy řízeného systému spočítá optimální nastavení parametrů regulátoru a v průběhu řízení se podle aktuálního stavu mezi těmito hodnotami přechází. Základní výhodou je, že s příslušnou sadou parametrů se pracuje pouze v okolí stavu, pro které byly spočítány. To umožňuje nahradit nelineární model v okolí daného stavu modelem linearizovaným.

Použitý regulátor vycházel z prediktivního řízení – Model Predictive Control (MPC), které dovoluje jednoduše zahrnout znalost budoucího průběhu žádané teploty.

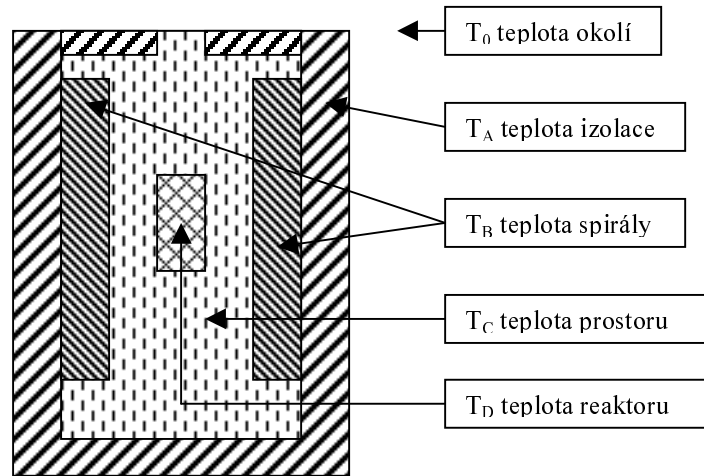
## 3. Matematický model

Chování pece s reaktorem bylo aproximováno náhradním zapojením 4 kapacit tak, jak je ukázáno na obrázku 1. Stavové veličiny jsou střední teploty jednotlivých kapacit. Vzhledem k tomu, že nejsou známy koeficienty prostupu tepla ani přestupní plochy, byly jako neznámé parametry určovány výsledné součiny fyzikálních parametrů. Tuto soustavu lze na základě tepelné bilance popsat soustavou čtyř lineárních diferenciálních rovnic s deseti neznámými parametry. Při popisu tímto lineárním modelem však nebylo možné nalézt takové hodnoty parametrů, který by zajistily rozumnou shodu s naměřenými hodnotami. Teprve po zavedení dále popsaných nelinearit (další tři parametry)

bylo dosaženo lepší shody (hodnota kritéria shody – směrodatná odchylka rozdílu průběhů se zmenšila cca 4x).

Nelinearity jsou do původně lineárního modelu zahrnuty jednak v závislosti odporu topné spirály na její teplotě (parametr  $\beta$ ) a jednak aproximací přenosu tepla mezi spirálou a reaktorem vyzařováním (závislost na čtvrté mocnině teploty spirály, parametry  $\gamma_1$  a  $\gamma_2$ ). Pro nulové hodnoty těchto tří parametrů model přejde na původní lineární závislost.

$\alpha$	koeficient přestupu tepla
$\rho$	hustota
$c$	měrná tepelná kapacita
$S$	přestupní plocha
$E$	příkon
$P_A = \rho_A V_A c_A$	$P_B = \rho_B V_B c_B$
$P_C = \rho_C V_C c_C$	$P_D = \rho_D V_D c_D$
$a_{AB} = \alpha_{AB} S_{AB}$	$a_{AC} = \alpha_{AC} S_{AC}$
$a_{A0} = \alpha_{A0} S_{A0}$	$a_{BC} = \alpha_{BC} S_{BC}$
$a_{CO} = \alpha_{CO} S_{CO}$	$a_{CD} = \alpha_{CD} S_{CD}$



Obrázek 1 Náhradní schéma pece

Matematický model je pak tvořen následující soustavou čtyř diferenciálních rovnic:

$$\begin{aligned} \frac{dT_A}{dt} &= \frac{1}{P_A} [a_{AB}(T_B - T_A) + a_{AC}(T_C - T_A) - a_{AO}(T_A - T_o)] \\ \frac{dT_B}{dt} &= \frac{E_0(t)}{P_B(1 + \beta T_B)} - \frac{1}{P_B} [a_{AB}(T_B - T_A) + a_{BC}(T_B - T_C) + (\gamma_1 + \gamma_2 T_B^4)(T_B - T_D)] \\ \frac{dT_C}{dt} &= \frac{1}{P_C} [a_{BC}(T_B - T_C) - a_{AC}(T_C - T_A) - a_{CD}(T_C - T_D) - a_{CO}(T_C - T_o)] \\ \frac{dT_D}{dt} &= \frac{1}{P_D} [a_{CD}(T_C - T_D) + (\gamma_1 + \gamma_2 T_B^4)(T_B - T_D)] \end{aligned}$$

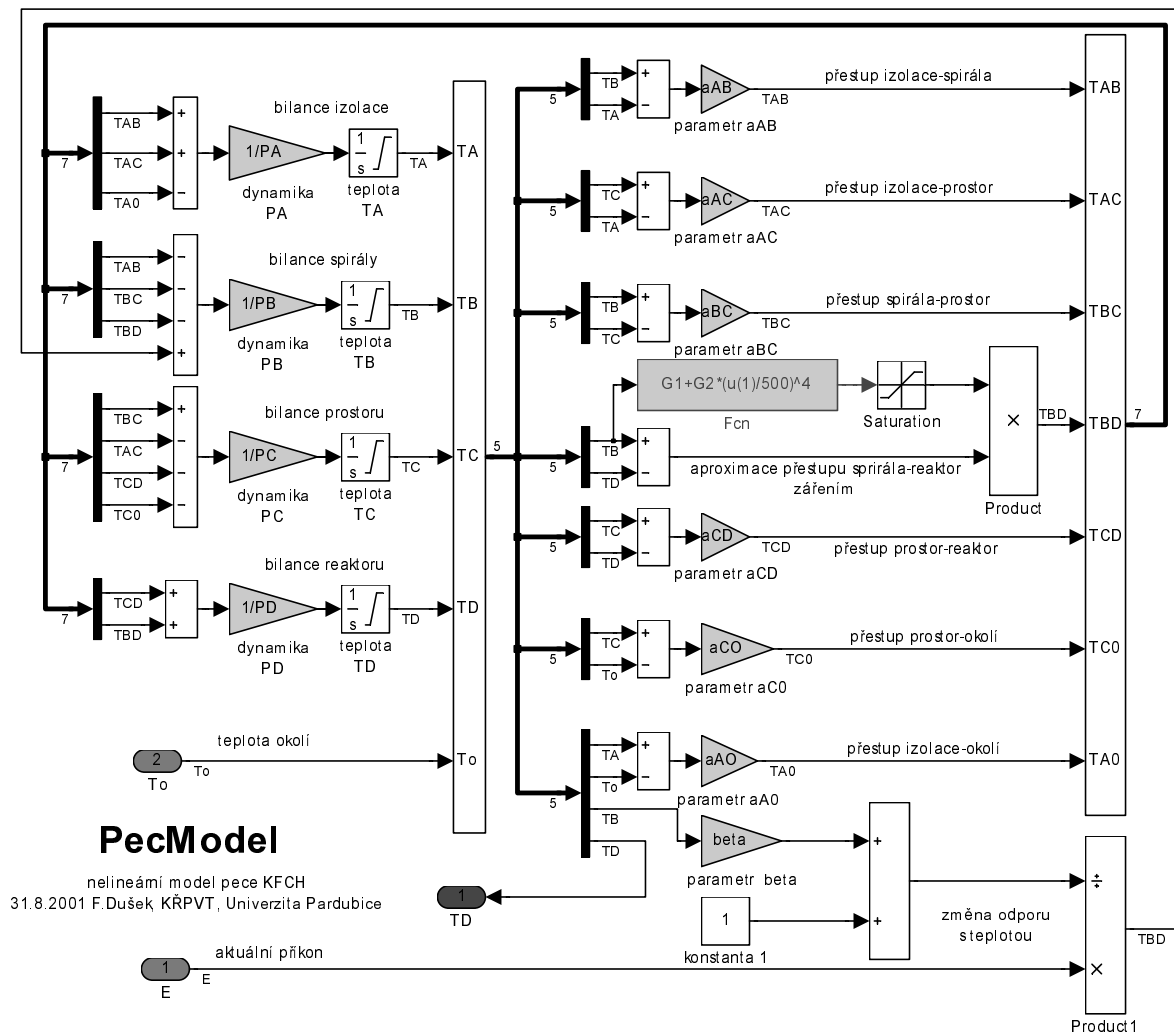
Odpovídající model v SIMULINKu (*PecModel.mdl*) ve formě subsystému je zobrazen na obrázku 2. Hodnoty parametrů modelu jsou přebírány z pracovního prostoru MATLABu (workspace), z proměnných jejichž název je použit jako parametr v příslušném modulu. Před prvním spuštěním simulace je tedy nutné v pracovním prostoru MATLABu tyto proměnné vytvořit a naplnit hodnotami.

#### 4. Určení parametrů modelu

Parametry modelu byly určeny experimentální identifikací. Standardní optimalizací byly vyhledány takové hodnoty třinácti parametrů modelu, které minimalizovaly směrodatnou odchylku rozdílu skutečné změřené teploty reaktoru a teploty  $T_D$  modelu pro daný časový průběh příkonu.

Byla využita standardní funkce **fminsearch** pro vyhledání minima funkce více proměnných. Funkce pro výpočet hodnoty směrodatné odchylky pak využívá možnost spuštění simulačního modelu pro určení výstupu modelu teploty pece pomocí funkce **sim**. Model pece (*Pec.mdl*) byl doplněn o načtení průběhu příkonu a uložení výsledné teploty do pracovního prostoru MATLABu pomocí bloků **FromWorkspace** a **ToWorkspace**. Příslušný model v SIMULINKu je na obrázku 3.

Určité komplikace způsobuje volání simulačního modelu z funkce. Parametry simulačního modelu se berou ze základního pracovního prostoru MATLABu a výsledek se uloží do lokálního pracovního prostoru funkce, ze které je simulační model volán. Je tedy nutné z funkce volající simulační model změnit hodnoty proměnných v základním prostoru. Nejjednodušší řešení je použití globálních proměnných. Toto propojení vybraných proměnných je vidět ve výpisech souborů *PecSim.m* a *PecSimKr.m* (příkaz **global**) uvedených v dodatku A.



Celé řešení úlohy experimentální identifikace parametrů nelineárního modelu na základě

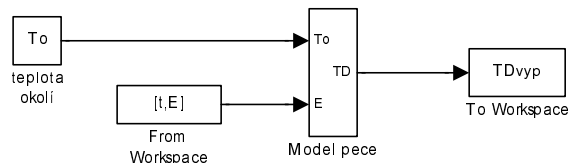
**Obrázek 2** SIMULINKový model pece ve formě subsystému

naměřených dat pak představuje jeden model v SIMULINKu (obr. 3) a dva soubory MATLABu. (*PecSim.m* a *PecSimKr.m*).

**Poznámka:** Tento způsob nalezení parametrů modelu samozřejmě nezaručuje, že jsou nalezeny skutečně fyzikálně správné hodnoty parametrů. Jsou nalezeny takové hodnoty parametrů, které zajišťují minimální hodnotu kritéria tj. z hlediska kritéria nejlepší shodu výstupu modelu s naměřenými daty pro daný konkrétní průběh vstupní veličiny. Neobsahují-li data dostatek informace, případně jsou zatíženy chybou měření nebo není dobrá shoda struktury modelu se skutečností, mohou vyjít hodnoty parametrů fyzikálně nereálné (např. záporné hodnoty koeficientu přestupu tepla atd.). V tomto konkrétním případě potřebujeme model, který co nejlépe aproximuje dynamický průběh teploty pece a nikoliv určit skutečné hodnoty parametrů. Proto je možné uvedený postup použít s vědomím, že do konkrétních hodnot parametrů jsou promítnuty i chyby vyplývající z neshody použité struktury se skutečným zařízením. Výhodou modelu, který má fyzikální podstatu, je pak to, že umožňuje určitou jeho verifikaci. Lze totiž odhadnout, byť velmi přibližně, rozsah ve kterém se mohou parametry pohybovat. Potom případné neobvyklé hodnoty identifikovaných parametrů signalizují, že buď není něco v pořádku s naměřenými daty nebo neshoda zvolené struktury modelu se skutečností je příliš velká.

### Experimentální identifikace

simulační model pece KFCH pro získání výstupu modelu pro zadaný průběh vstupního příkonu  
 31.8.2001 F. Dušek KRPVT, Univerzita Pardubice



**Obrázek 3** Simulační schéma

## 5. Linearizace nelineárního modelu

Na teoretické úrovni není linearizace nelineárního modelu problém. Obecný nelineární model o  $n_x$  stavech,  $n_u$  vstupech a  $n_y$  výstupech tvořený soustavou  $n_x$  nelineárních diferenciálních a  $n_y$  algebraických rovnic lze zapsat ve vektorovém tvaru jako

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad \mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u})$$

kde  $\mathbf{u}$  vektor  $n_u$  vstupních veličin  
 $\mathbf{y}$  vektor  $n_y$  výstupních veličin  
 $\mathbf{x}$  vektor  $n_x$  stavových veličin  
 $\mathbf{f}$  vektor  $n_x$  funkcí soustavy diferenciálních rovnic  
 $\mathbf{g}$  vektor  $n_y$  funkcí soustavy algebraických hodnot

Linearizace v bodě  $[\mathbf{x}_0=\mathbf{x}(t_0), \mathbf{u}_0=\mathbf{u}(t_0)]$  je pak popsána jako

$$\frac{d\mathbf{x}_l}{dt} = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}_0, \mathbf{u}_0}}_{\mathbf{A}} (\mathbf{x}_l - \mathbf{x}_0) + \underbrace{\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \Big|_{\mathbf{x}_0, \mathbf{u}_0}}_{\mathbf{B}} (\mathbf{u} - \mathbf{u}_0) = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \mathbf{A}(\mathbf{x}_l - \mathbf{x}_0) + \mathbf{B}(\mathbf{u} - \mathbf{u}_0)$$

$$\mathbf{y}_l = \underbrace{\mathbf{g}(\mathbf{x}_0, \mathbf{u}_0)}_{\mathbf{y}(t_0)} + \underbrace{\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \Big|_{\mathbf{x}_0, \mathbf{u}_0}}_{\mathbf{C}} (\mathbf{x}_l - \mathbf{x}_0) + \underbrace{\frac{\partial \mathbf{g}}{\partial \mathbf{u}} \Big|_{\mathbf{x}_0, \mathbf{u}_0}}_{\mathbf{D}} (\mathbf{u} - \mathbf{u}_0) = \mathbf{g}(\mathbf{x}_0, \mathbf{u}_0) + \mathbf{C}(\mathbf{x}_l - \mathbf{x}_0) + \mathbf{D}(\mathbf{u} - \mathbf{u}_0)$$

Absolutní člen první rovnice je nulový v případě, že zvolený bod linearizace odpovídá tzv. ustálenému stavu - změny (derivace) všech stavových veličin jsou nulové tj. platí rovnice  $\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0)=0$ . Absolutní člen druhé rovnice lze převést na levou stranu a dále pracovat s odchylkou výstupu od hodnoty  $\mathbf{y}_0=\mathbf{y}(t_0)=\mathbf{g}(\mathbf{x}_0, \mathbf{u}_0)$ . Provedeme-li tedy linearizace v bodě ustáleného stavu je výsledkem standardní stavový model v odchylkovém tvaru.

$$\frac{d\Delta \mathbf{x}_l}{dt} = \mathbf{A}\Delta \mathbf{x}_l + \mathbf{B}\Delta \mathbf{u} \quad \Delta \mathbf{x}_l = (\mathbf{x}_l - \mathbf{x}_0) \quad \Delta \mathbf{u} = (\mathbf{u} - \mathbf{u}_0)$$

$$\Delta \mathbf{y}_l = \mathbf{C}\Delta \mathbf{x}_l + \mathbf{D}\Delta \mathbf{u} \quad \Delta \mathbf{y}_l = \mathbf{y}_l - \underbrace{\mathbf{g}(\mathbf{x}_0, \mathbf{u}_0)}_{\mathbf{y}_0}$$

Matice parciálních derivací  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  a  $\mathbf{D}$  představují jakobiány příslušných vektorových funkcí

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}_0, \mathbf{u}_0} = \mathbf{A} = \begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_{n_x}} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial x_1} & \frac{\partial f_{n_x}}{\partial x_2} & \dots & \frac{\partial f_{n_x}}{\partial x_{n_x}} \end{vmatrix}_{\mathbf{x}_0, \mathbf{u}_0}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \Big|_{\mathbf{x}_0, \mathbf{u}_0} = \mathbf{B} = \begin{vmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \dots & \frac{\partial f_1}{\partial u_{n_u}} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \dots & \frac{\partial f_2}{\partial u_{n_u}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial u_1} & \frac{\partial f_{n_x}}{\partial u_2} & \dots & \frac{\partial f_{n_x}}{\partial u_{n_u}} \end{vmatrix}_{\mathbf{x}_0, \mathbf{u}_0}$$

$$\left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x}_0, \mathbf{u}_0} = \mathbf{C} = \begin{vmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \dots & \frac{\partial g_1}{\partial x_{n_x}} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \dots & \frac{\partial g_2}{\partial x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{n_x}}{\partial x_1} & \frac{\partial g_{n_x}}{\partial x_2} & \dots & \frac{\partial g_{n_x}}{\partial x_{n_x}} \end{vmatrix}_{\mathbf{x}_0, \mathbf{u}_0} \quad \left. \frac{\partial \mathbf{g}}{\partial \mathbf{u}} \right|_{\mathbf{x}_0, \mathbf{u}_0} = \mathbf{D} = \begin{vmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} & \dots & \frac{\partial g_1}{\partial u_{n_u}} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} & \dots & \frac{\partial g_2}{\partial u_{n_u}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{n_x}}{\partial u_1} & \frac{\partial g_{n_x}}{\partial u_2} & \dots & \frac{\partial g_{n_x}}{\partial u_{n_u}} \end{vmatrix}_{\mathbf{x}_0, \mathbf{u}_0}$$

Teoreticky není nalezení matic **A**, **B**, **C** a **D** problém. Prakticky však jde o poměrně pracnou záležitost. V našem případě ji navíc komplikuje požadavek stanovit ustálený stav takový, aby splňoval požadavky na dodržení zadané hodnoty výstupu a jedné ze dvou hodnot vstupu.

Řešení linearizace nelineárního modelu se tedy rozpadá na dvě úlohy. První úlohou je nalezení bodu linearizace (určení stavu  $\mathbf{x}_0$  a některých vstupů z vektoru  $\mathbf{u}_0$ ) splňujícího zadané podmínky (některé hodnoty vektorů  $\mathbf{y}_0$  a  $\mathbf{u}_0$ ) tj. řešit soustavu algebraických rovnic

$$\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) = 0 \quad a \quad \mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0, \mathbf{u}_0)$$

$$pro \quad u_{0,1..n} = konst \quad a \quad y_{0,1..m} = konst$$

Tato úloha může být z hlediska hledaných neznámých obecně nedourčená i přeúčtená. V našem případě čtyř rovnic pro šest proměnných (čtyři stavové -  $T_A, T_B, T_C, T_D$  - z nichž jedna -  $T_D$  - je přímo výstupní a dvou vstupních -  $E_0, T_0$  -) se hledaly hodnoty teplot  $T_A, T_B, T_C$  a příkonu  $E_0$  takové, aby bylo dosaženo zadané teploty  $T_D$  při zadané teplotě okolí  $T_0$ . Máme-li nelineární model ve formě SIMULINKového modelu, řeší tuto úlohu funkce MATLABu **trim**. Toto vyhledávání není triviální záležitostí a při nevhodně zadaných podmínkách nemusí mít řešení, případně může být řešení několik v závislosti na zadaných startovacích hodnotách určovaných neznámých.

Druhou úlohou je vlastní linearizace tj. určení matic **A**, **B**, **C** a **D** (hodnot jakobiánů) respektive stavového modelu pro určený bod  $[\mathbf{y}_0, \mathbf{x}_0, \mathbf{u}_0]$ . Tuto úlohu řeší funkce MATLABu **linmod**.

Řešení obou úloh pro zvolené hodnoty výstupní teploty je ukázáno v souboru *LinMod.m*, který je uveden v dodatku A.

## 6. Závěr

V článku je popsán postup získání linearizovaného modelu (soustavy lineárních diferenciálních a algebraických rovnic) nahrazujících chování původního nelineárního modelu (soustavy nelineárních diferenciálních a algebraických rovnic) v okolí vybraného bodu. Je naznačen i postup získání neznámých parametrů původního nelineárního modelu z experimentálně získaných dat. V tomto postupu je intenzivně využíváno možností, které poskytuje programový systém MATLAB a SIMULINK pro snížení pracnosti řešení.

Kontaktní informace:

doc. ing. František Dušek, CSc.  
frantisek.dusek@upce.cz  
tel.: 040/6037125

ing. Daniel Honc  
daniel.honc@upce.cz  
tel.: 040/6037107

Katedra řízení procesů a výpočetní techniky (KŘPVT)  
Fakulta chemickotechnologická (FCHT)  
Univerzita Pardubice  
nám. Čs. legií 565, 53210 Pardubice

## Dodatek A – výpisy souborů

### ***PecSim.m***

Skript pro zpracování dat uložených v souboru merdata.txt (čas,příkon,teplota). Jsou vyhledány hodnoty 13 proměnných, které minimalizují směrodatnou odchylku rozdílu mezi naměřeným průběhem teploty reaktoru a průběhem vypočteným z modelu pece. Výsledek je uložen do souboru *pecpar.mat*. Výpočet kritéria pro zadané parametry provádí uživatelská m-funkce *PecSimKr.m*.

```
% skript pro zpracování dat PEC

load merdata.txt

N=2200;           % prvních N dat
t=merdata(1:N,1); % cas v sec
E=merdata(1:N,2); % příkon = vstup soustavy
TD=merdata(1:N,3); % teplota reaktoru = výstup soustavy
To=TD(1);        % teplota okolí

global aAO aAB aAC aBC aCO aCD PA PB PC PD beta G1 G2
% startovací hodnoty parametrů
aAO=0.099;aAB=0.3956;aAC=0.0476; aBC=3.819; aCD=0.000183;
aCO=0.016; G1=0.00029;G2=0.00063;beta=0.0019;
PA =191.7; PB=108.9; PC=0.1291; PD=0.013;
% vektor parametrů - počáteční odhad
p0=[aAO,aAB,aAC,aBC,aCO,aCD,G1,G2,PA,PB,PC,PD,beta];
% vyhledání minima fce více proměnných
o=optimset; optimset(o,'MaxFunEvals',100);
p=fminsearch('PecSimKr',p0,o,TD,[t(1),t(end)]);
% uložení výsledku pro další použití
save pecpar p t E TD
```

### ***PecSimKr.m***

Uživatelská m-funkce vyhodnocující hodnotu směrodatné odchylky naměřeného průběhu od vypočteného průběhu pomocí volání modelu SIMULINKu *pec.mdl*. Je volána z funkce *fminsearch* ze skriptu *PecSim.m*.

```
function Kr=PecSimKr(p,y,Tspan)
% výpočet kritéria pro dané parametry
% p=[aA0,aAB,aAC,aBC,aCO,aCD,G1,G2,PA,PB,PC,PD,beta]
% y ... průběh měřeného výstupu soustavy
% Tspan časový interval výpočtu

global aA0 aAB aAC aBC aCO aCD PA PB PC PD beta G1 G2

% přenos hodnot do základního pracovního prostoru
aA0=p(1);aAB=p(2); aAC=p(3);aBC=p(4);aCO=p(5);aCD=p(6);
PA= p(9); PB=p(10); PC=p(11);PD=p(12);
G1= p(7); G2=p(8);beta=p(13);

sim('pec',Tspan); % spuštění výpočtu modelu Pec.mdl
% výsledek uložen do TDvyp v prac. prostoru funkce

% výpočet směrodatné odchylky
Kr=sqrt((y-TDvyp)'*(y-TDvyp)/length(y));
```

### ***PecLin.m***

Skript počítající linearizovaný model nelineárního modelu (*PecModel.mdl*) v SIMULINKu v několika ustálených stavech. Parametry SIMULINKového modelu jsou vzaty ze souboru *pecpar.mat* a výsledné

```
% určení linearizovaných modelů pro vybrané ustálené stavy
%
load pecpar
% parametry z identifikace
aAO=p(1);aAB=p(2); aAC=p(3);
aBC=p(4);aCO=p(5); aCD=p(6);
PA=p(9); PB=p(10); PC=p(11); PD=p(12);
G1=p(7); G2=p(8);beta=p(13);
% okolní teplota
To=TD(1);

Y0=[100 200 300 400 600 800]; % hodnota výstupu v ustál. stavech
PocBodu=length(Y0); % počet ustálených stavů
X0=100*ones(4,1); U0=[100; To]; % poč.odhad X0, U0 [příkon, ok.tepl.]

for i = 1:PocBodu;
% hledání ustáleného stavu
% [X,U,Y,DX]=TRIM('SYS',X0,U0,Y0,IX,IU,IY) zafixuje X, U a Y na
hodnoty X0(IX), U0(IU) a Y0(IY)
[X,U,Y,DX]=trim('PecModel',X0,U0,Y0(i),[],[2],[1]);
% získání lineárního modelu
% [A,B,C,D]=LINMOD('SYS',X,U) vektor stavu X avstupu U
[A,B,C,D]=linmod('PecModel',X,U);
% LinModel je strukturovaný vektor výsledků s položkami:
% model, vstupy, vystupy, stav
% napr. LinModel(1).model je stavový model pro první ustálený stav
% LinModel(end).vystupy jsou vystupy u posledního ustál. stavu
LinModel(i).model=ss(A,B,C,D);
LinModel(i).vstupy=U;
LinModel(i).vystupy=Y;
LinModel(i).stav=X;
end
save PecParLin LinModel
```

linearizované modely jsou uloženy do souboru *PecParLin.mdl* ve struktuře *LinModel*.