

APLIKÁCIA FUZZY ALGORITMOV RIADENIA V PROSTREDÍ MATLAB-SIMULINK

*Z. Dideková, ** Š. Kozák

*Ústav riadenia a priemyselnej informatiky, Fakulta elektrotechniky a informatiky,

**Ústav aplikovanej informatiky, Fakulta informatiky a informačných technológií,
Slovenská technická univerzita v Bratislave, Slovenská republika

klúčové slová: fuzzy, fuzzy logika, Hammersteinove modely, Matlab-Simulink aplikácie

1. Úvod

Príspevok pojednáva o využití programového systému Matlab-Simulink pre modelovanie a riadenie silno nelineárnych systémov. Nelineárny systém je modelovaný tzv. fuzzy Hammersteinovým modelom a pre riadenie je použitý prediktívny algoritmus GPC. Použitý matematický aparát modelovania a riadenia vo Fuzzy Toolboxe bol overený na reálnom príklade riadenia pH faktora.

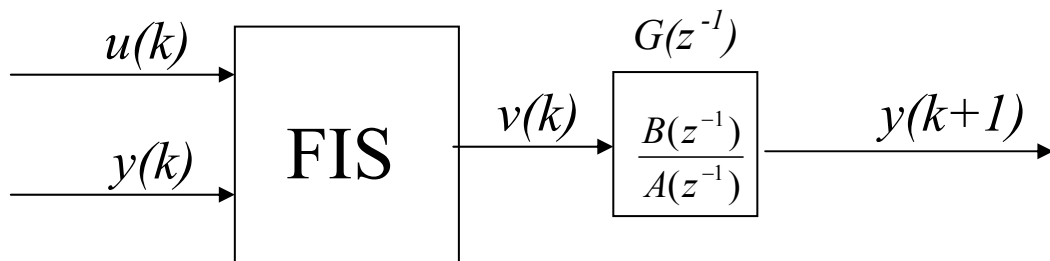
Oblasť automatického riadenia (AC) je stará viac ako 60 rokov. Princípy AC prenikajú do takých rozdielnych oblastí ako je biológia, medicína, bankovníctvo, ekonómia atď. a s ich aplikáciami sa môžeme stretnúť prakticky všade v našom každodennom živote.

Niekoľko desaťročí sú značne využívané konvenčné štandardné stratégie riadenia. Veľká väčšina automatických riadiacich slučiek v priemysle procesov (90%) sa stále spolieha na rôzne formy všadeprítomných PID regulátorov. [1]

Obvyklé metódy riadenia nie sú použiteľné pre silno nelineárne procesy. V posledných rokoch bola výskumná aktivita zameraná na riešenie úloh modelovania a riadenia nelineárnych systémov použitím metód výpočtovej inteligencie. Fuzzy množiny patria do skupiny tých efektívnych metód výpočtovej inteligencie, kde je možné pri modelovaní nelineárnych procesov využiť expertné metódy. Umelá neurónová sieť (ANN), genetický algoritmus (GA), fuzzy logika, hybridné a adaptívne soft metódy umožňujú transformovať skúsenosti experta do návrhu riadiacich algoritmov.

Hammersteinov model je jeden z najjednoduchších a najpopulárnejších členov rodiny blokovo-orientovaných nelineárnych dynamických modelov. Štruktúra Hammersteinovho modelu pozostáva zo statickej nelinearity $g(\cdot)$, spojenej v kaskáde s lineárnym dynamickým modelom definovaným prenosovou funkciou $H(z)$. [3]

Fuzzy Hammersteinov model (FH model) pozostáva zo statického fuzzy modelu spojeného do série s lineárnym dynamickým modelom. Statická nelinearita $f(\cdot)$ je reprezentovaná funkciou $v = f(u)$. V predložennom príspevku je $f(\cdot)$ funkcia reprezentovaná Takagi-Sugeno fuzzy modelom nultého rádu. Lineárna časť dynamického modelu je reprezentovaná dynamickým modelom typu ARX. [4]



Obr. 1 Fuzzy Hammersteinov model

2. Formulácia problematiky

Identifikácia FH modelu

FH model je vyjadrený pomocou vzťahu:

$$y(k+1) = \sum_{i=1}^{n_a} a_i y(k-i+1) + \sum_{i=1}^{n_b} b_i \frac{\sum_{j=1}^{N_R} \beta_j(u(k-i-n_d+1), y(k-i-n_d+1)) d_j}{\sum_{j=1}^{N_R} \beta_j(u(k-i-n_d+1), y(k-i-n_d+1))} \quad (1)$$

Váha $0 \leq \beta_{(i_u, i_y)}(u, y) \leq 1$, hodnota (i_u, i_y) -tého pravidla sa vypočíta ako:

$$\beta_{(i_u, i_y)}(u, y) = A_{(u, i_u)}(u) \times A_{(y, i_y)}(y) \quad (2)$$

kde $A_{(u, i_u)}$, $A_{(y, i_y)}$ sú Gaussove funkcie. N_R označuje počet pravidiel fuzzy modelu, n_a je rád menovateľa a n_b rád čitateľa ARX modelu.

Parametre a_i a b_i , patriace lineárnemu dynamickému modelu, budú nazývané „lineárne parametre“, zatiaľ čo parametre d_j patriace fuzzy modelu budú nazývané „nelineárne parametre“.

V príspevku sa zaoberame jedнокrokovým iteračným riešením identifikácie FH modelu. Trieda jedнокrokových, iteračných riešení zahŕňa techniky, ktoré striedavo vylepšujú odhad dynamickej lineárnej časti a statickej nelinearity. Pre identifikáciu sa používajú iba dynamické dáta. Použitá metóda je modifikáciou algoritmu navrhnutého Narendrom a Gallmanom pre identifikáciu Hammersteinovho modelu s polynomiálnou statickou nelinearitou. Parametre modelu sa získajú oddelením odhadu lineárnej dynamiky od odhadu statickej nelineárnej časti. Na začiatku nie je známe nič z týchto častí, k riešeniu sa dospeje použitím iteračného identifikačného algoritmu.

Pre jednoduchosť je možné predpokladať, že podmienková časť fuzzy modelu (fuzzy množiny) je navrhnutá manuálne, na základe apriórnych informácií. Možno tiež využiť numerické optimalizačné techniky, ako napr. fuzzy clustering. Identifikačný algoritmus potom určí parametre a_i , b_i a d_j . Je to jasne nelineárny optimalizačný problém. Pre identifikáciu možno použiť on-line alebo off-line prístup. Kvôli jednoduchošti použijeme iteračnú off-line procedúru.

Najskôr možno predpokladať, že parametre a_i , b_i lineárneho dynamického modelu sú známe. Potom parametre $\bar{d} = [d_1, \dots, d_{N_R}]$ nelineárnej časti môžu byť odhadnuté riešením regresného problému:

$$\bar{y}_d = \bar{\phi}_d \bar{d} + \bar{\varepsilon} \quad (3)$$

kde $\bar{\varepsilon}$ označuje chybu modelovania s nulovou strednou hodnotou a normálnym rozdelením. Pre N dátových párov, vektor \bar{y}_d a matica $\bar{\phi}_d$ sú dané vzťahmi:

$$\bar{y}_d = \begin{bmatrix} y_d(1) \\ y_d(2) \\ \vdots \\ y_d(N) \end{bmatrix}, \quad \bar{\phi}_d = \begin{bmatrix} \bar{\phi}_d(1) \\ \bar{\phi}_d(2) \\ \vdots \\ \bar{\phi}_d(N) \end{bmatrix}, \quad (4)$$

kde:

$$y_d(k) = y(k) - \sum_{i=1}^{n_a} a_i y(k-i),$$

$$\bar{\phi}_d(k) = \left[\sum_{i=1}^{n_b} b_i \beta_i(u(k-i-n_d)), \dots, \sum_{i=1}^{n_b} b_i \beta_{N_R}(u(k-i-n_d)) \right] / \sum_{i=1}^{N_R} \beta_i(u(k-i-n_d)) \quad (5)$$

Riešenie sa získa metódou najmenších štvorcov:

$$\bar{d} = \left[\bar{\phi}_d^T \bar{\phi}_d \right]^{-1} (\bar{\phi}_d)^T \bar{y}_d. \quad (6)$$

Parametre lineárneho dynamického modelu $\bar{\theta}_l = [a_1, \dots, a_{n_y}, b_1, \dots, b_{n_u}]^T$ môžu byť odhadnuté riešením nasledovného regresného problému:

$$\bar{y}_l = \bar{\phi}_l \bar{\theta}_l + \bar{\varepsilon}, \quad (7)$$

kde:

$$\bar{y}_l = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}, \quad \bar{\phi}_l = \begin{bmatrix} \bar{\phi}_l(1) \\ \bar{\phi}_l(2) \\ \vdots \\ \bar{\phi}_l(N) \end{bmatrix}, \quad (8)$$

a:

$$\bar{\phi}_l(k) = [y(k-1), \dots, y(k-n_a), \frac{\sum_{j=1}^{N_R} \beta_j(u(k-1-n_d)) d_j}{\sum_{j=1}^{N_R} \beta_j(u(k-1-n_d))}, \dots, \frac{\sum_{j=1}^{N_R} \beta_j(u(k-n_b-n_d)) d_j}{\sum_{j=1}^{N_R} \beta_j(u(k-n_b-n_d))}] \quad (9)$$

Odhad lineárnych parametrov metódou najmenších štvorcov sa vypočíta pomocou vzťahu:

$$\bar{\theta}_l = \left[\bar{\phi}_l^T \bar{\phi}_l \right]^{-1} \bar{\phi}_l^T \bar{y}_l. \quad (10)$$

Nelineárne parametre \bar{d} sú potom odhadované znovu použitím (6) a celá procedúra sa iteračne opakuje.

Keďže statické zosilnenie FH modelu je určené aj statickou nelinearitou, aj zosilnením lineárnej časti, model je redundandný. Takže táto metóda môže divergovať.

Jednotkové zosilnenie identifikovaného lineárneho modelu je zaistené použitím viazaného kvadratického programovania (QP) namiesto (10):

$$\min_{\bar{\theta}_l} \left\{ \frac{1}{2} \bar{\theta}_l^T \bar{H} \bar{\theta}_l + \bar{c}^T \bar{\theta}_l \right\}, \quad (11)$$

s $\bar{H} = 2 \bar{\phi}_l^T \bar{\phi}_l$ a $\bar{c} = -2 (\bar{\phi}_l)^T \bar{y}_l$. Obmedzenie $\bar{\theta}_l$ môže byť napísané ako:

$$[1, 1, \dots, 1] \bar{\theta}_l = 1. \quad (12)$$

Iteračný algoritmus sa zastaví, ak je pri oboch, $\bar{\theta}_l$ aj \bar{d} , splnená podmienka konvergencie.

Výraz „konvergencia“ tu znamená, že norma rozdielu dvoch parametrov medzi dvoma po sebe idúcimi iteráciami je menšia ako preddefinovaná prahová úroveň. Hore uvedený algoritmus je výpočtovo náročný, pretože vyžaduje riešenie jedného problému metódou najmenších štvorcov a jedného kvadratickým programovaním v každej iterácii. Navyiac, keďže ide o off-line algoritmus, ak sú platné nové vstupno-výstupné dáta, celý algoritmus sa musí reštartovať. [4]

Riadenie nelineárneho systému pomocou FH modelu a GPC algoritmu riadenia

Pre riadenie nelineárneho systému je použitý jeho FH model a prediktívny algoritmus GPC. Keďže algoritmus GPC využíva ARX model systému, je nutné pri každom výpočte akčného zásahu prepočítať FH model systému na ARX model. Využíva sa pri tom linearizácia FH modelu v pracovnom bode.

Linearizáciou rovnice FH modelu:

$$y(k+1) = \sum_{i=1}^{n_a} a_i y(k-i+1) + \sum_{i=1}^{n_b} b_i \frac{\sum_{j=1}^{N_R} \beta_j(u(k-i+1), y(k-i+1)) d_j}{\sum_{j=1}^{N_R} \beta_j(u(k-i+1), y(k-i+1))} \quad (13)$$

v pracovnom bode $(x, y(x)) = ((u(k), y(k)), y(k+1), u(k), y(k))$ sa dospeje k lineárnej rovnici, ktorá zastupuje ARX model:

$$y(k+1) = \sum_{i=1}^{n_b} dyu(k-i+1) \Delta u(k-i+1) + (1+dy y(k)) y(k) + \sum_{i=1}^{n_{a2}-1} (dy y(k-i) - dy y(k-i+1)) y(k-i) - dy y(k-n_{a2}+1) y(k-n_{a2}) \quad (14)$$

$$\begin{aligned} n_{a2} &= n_a + 1 & \text{ak } n_a > n_b \\ &= n_b + 1 & \text{ak } n_a \leq n_b \end{aligned}$$

$$dyu(k-i) = b_{i+1} \left. \frac{\partial v(k, u(k), y(k))}{\partial u(k)} \right|_{\substack{u(k)=u(k-i-1) \\ y(k)=y(k-i)}} \quad \text{pre } i \in \langle 0; n_b - 1 \rangle \quad (15)$$

$$dy y(k-i) = a_{i+1} + b_{i+1} \left. \frac{\partial v(k, u(k), y(k))}{\partial y(k)} \right|_{\substack{u(k)=u(k-i-1) \\ y(k)=y(k-i)}} \quad \text{pre } i \in \langle 0; n_{a2} - 1 \rangle \quad (16)$$

$$v(k, u(k), y(k)) = \frac{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \beta_{ij}(u(k), y(k)) \cdot d_{ij}}{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \beta_{ij}(u(k), y(k))} \quad (17)$$

$$\beta_{ij}(u(k), y(k)) = e^{-\frac{(u(k)-c_{ui})^2}{2\text{sig}_{ui}^2}} \cdot e^{-\frac{(y(k)-c_{yj})^2}{2\text{sig}_{yj}^2}} \quad (18)$$

M_1 a M_2 predstavujú počet fuzzy množín pre jednotlivé vstupy $u(k)$ a $y(k)$ a c_{ui} , sig_{ui} , c_{yj} a sig_{yj} parametre jednotlivých fuzzy množín.

Ďalej sa pokračuje ako v klasickom prediktívnom algoritme GPC.

3. Overenie algoritmov modelovania a riadenia pre riadenie pH

Proces prebieha v nádrži, kde sa miešajú tri toky:

- kyslý (HCl) tok (q_1)
- vyrovnávací (KH_2PO_4 a Na_2HPO_4) tok (q_2) a
- zásaditý (NaOH) tok (q_3).

Výstupom z procesu je hodnota pH roztoku, ktorá môže byť regulovaná pôsobením na ventily.

Spojitély model sa dá získať za predpokladu: dokonalého miešania, konštantnej hustoty a úplnej rozpustnosti molekúl. Tento model môže byť opísaný nasledujúcimi rovnicami:

$$Ah \frac{dW_{a4}}{dt} = q_1(W_{a1} - W_{a4}) + q_2(W_{a2} - W_{a4}) + q_3(W_{a3} - W_{a4}) \quad (19)$$

$$Ah \frac{dW_{b4}}{dt} = q_1(W_{b1} - W_{b4}) + q_2(W_{b2} - W_{b4}) + q_3(W_{b3} - W_{b4}) \quad (20)$$

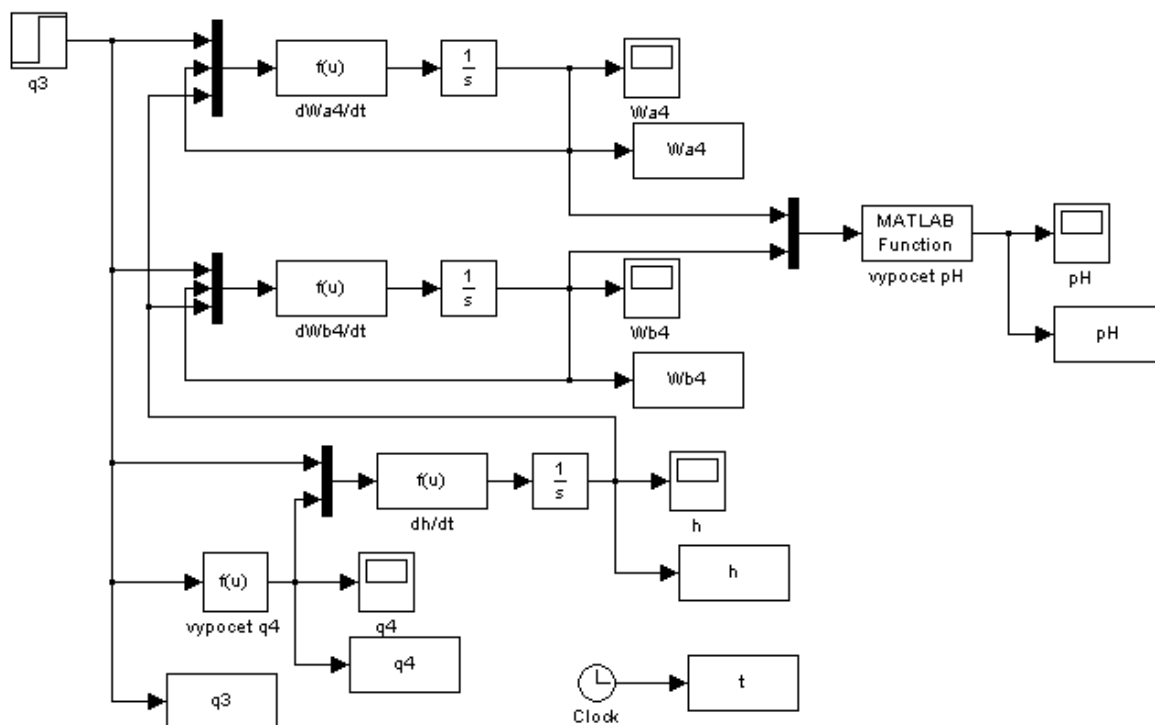
$$A \frac{dh}{dt} = q_1 + q_2 + q_3 - q_4 \quad (21)$$

$$W_{b4} \frac{K_{a4} / [H_3O^+] + 2K_{a4}K_{a5} / [H_3O^+]^2 + 3K_{a4}K_{a5}K_{a6} / [H_3O^+]^3}{1 + K_{a4} / [H_3O^+] + K_{a4}K_{a5} / [H_3O^+]^2 + K_{a4}K_{a5}K_{a6} / [H_3O^+]^3} + W_{a4} + \frac{K_w}{[H_3O^+]} - [H_3O^+] = 0 \quad (22)$$

$$pH = -\log[H_3O^+] \quad (23)$$

[5]

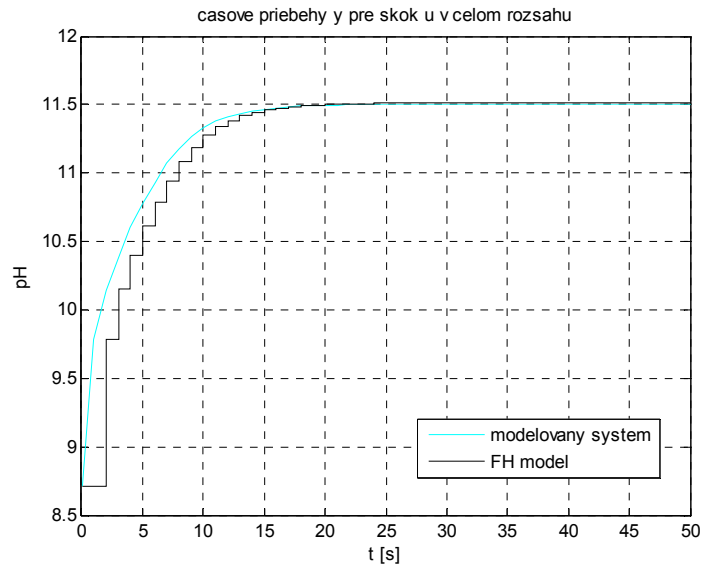
Schéma modelu procesu v Simulinku sa nachádza na obrázku Obr. 2.



Obr. 2 Schéma procesu neutralizácie pH v prostredí Matlab-Simulink

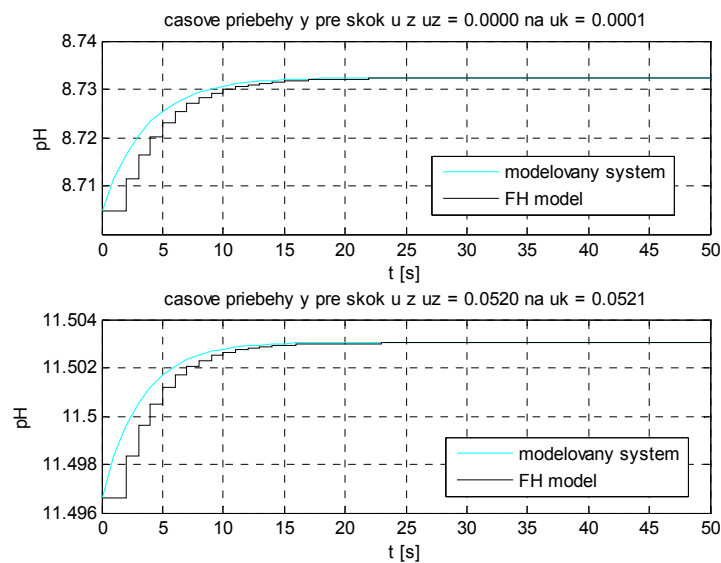
Modelovanie procesu

Na obrázku Obr. 3 sú porovnané časové priebehy výstupu y pre modelovaný systém a jeho FH model pre skok akčnej veličiny u (q_3) v celom rozsahu, v akom bol proces modelovaný z $u_z = 0,0000 \text{ l.s}^{-1}$ na $u_k = 0,0521 \text{ l.s}^{-1}$. Výstupná veličina y predstavuje hodnotu pH.



Obr. 3 Časové priebehy výstupu y pre modelovaný systém a jeho FH model pre skok akčnej veličiny u v celom rozsahu, v akom bol proces modelovaný

Na obrázku Obr. 4 sú porovnané časové priebehy výstupu y pre modelovaný systém a jeho FH model pre veľmi malé skoky akčnej veličiny u (q_3). V hornej časti obrázku sú zobrazené časové priebehy výstupu y pre skok u zo spodnej časti rozsahu u z $u_z = 0,0000 \text{ l.s}^{-1}$ na $u_k = 0,0001 \text{ l.s}^{-1}$. V dolnej časti obrázku sú zobrazené časové priebehy výstupu y pre skok u z hornej časti rozsahu u z $u_z = 0,0000 \text{ l.s}^{-1}$ na $u_k = 0,0001 \text{ l.s}^{-1}$.

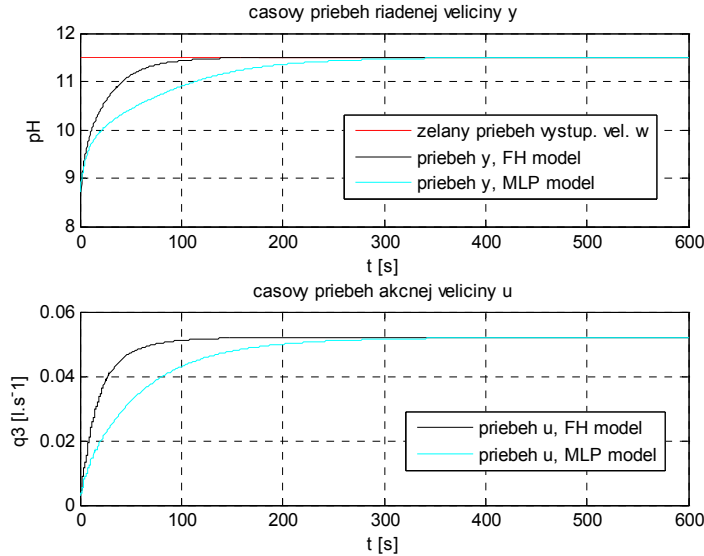


Obr. 4 Časové priebehy výstupu y pre modelovaný systém a jeho FH model pre veľmi malé skoky akčnej veličiny u

Časový priebeh výstupu z FH modelu sa oneskoruje za časovým výstupom z modelovaného systému o jednu periódu vzorkovania. Dynamika FH modelu zodpovedá dynamike modelovaného systému. Statická chyba modelu je takmer nulová (dosahuje hodnoty v absolútnej hodnote pod 2 %). Vytvorený FH model je vhodný na riadenie pomocou prediktívneho algoritmu GPC.

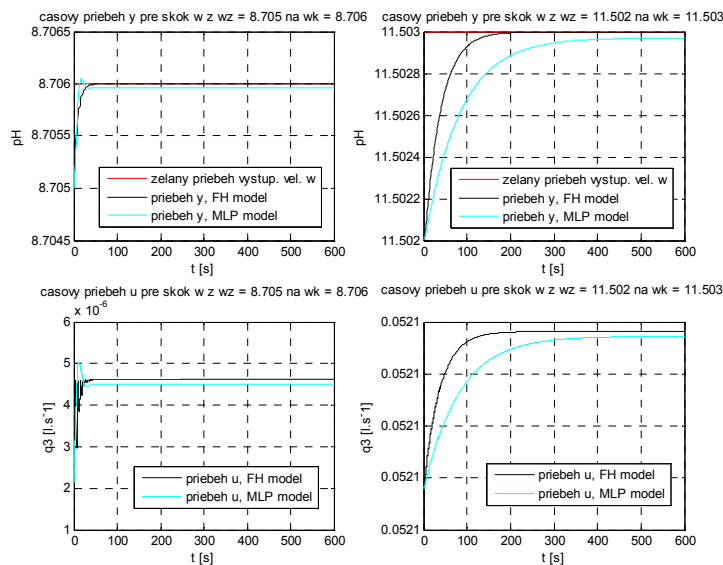
Porovnanie prediktívneho riadenia s FH modelom a prediktívneho riadenia s MLP modelom

Prediktívne riadenie s MLP modelom využíva ako model systému neurónovú sieť MLP (multi layer perceptron). Na obrázku Obr. 5 sú porovnané časové priebehy výstupu y a akčného zásahu u pri prediktívnom riadení s FH modelom a s MLP modelom pre skok referenčného signálu w v celom jeho rozsahu z $w_z = 8,705$ jednotiek pH na $w_k = 11,503$ jednotiek pH.



Obr. 5 Porovnanie časových priebehov výstupu y a akčného zásahu u pri prediktívnom riadení s FH modelom a s MLP modelom pre skok referenčného signálu w v celom jeho rozsahu

Na obrázku Obr. 6 sú porovnané časové priebehy výstupu y a akčného zásahu u pri prediktívnom riadení s FH modelom a s MLP modelom pre veľmi malé skoky referenčného signálu w . V hornej časti obrázku sú zobrazené časové priebehy výstupu y a v dolnej časti časové priebehy akčnej veličiny u pre jednotlivé skoky w . Na ľavej časti obrázku sú zobrazené časové priebehy y a u pre skok w z $w_z = 8,705$ jednotiek pH na $w_k = 8,706$ jednotiek pH. Na pravej časti obrázku sú zobrazené časové priebehy y a u pre skok w z $w_z = 11,502$ jednotiek pH na $w_k = 11,503$ jednotiek pH.



Obr. 6 Porovnanie časových priebehov výstupu y a akčného zásahu u pri prediktívnom riadení s FH modelom a s MLP modelom pre veľmi malé skoky referenčného signálu w

Obidva použité algoritmy riadenia zabezpečili stabilitu a vysokú kvalitu riadenia. Pri zabezpečení prerégulovania pod 5 % bola však doba regulácie riadenej veličiny s riadením s MLP modelom o niečo dlhšia. Pri prediktívnom riadení s MLP modelom sa tiež pre niektoré skoky objavila malá trvalá regulačná odchýlka v dôsledku chybajúcej integračnej zložky.

4. Záver

Navrhnutý prediktívny regulátor s FH modelom bol komplexne spracovaný a zovšeobecnený postup modelovania riadenia v prostredí Matlab-Simulink. Algoritmus dokáže zabezpečiť, aby regulovaná veličina s vysokou kvalitou sledovala časový priebeh želanej výstupnej veličiny pre každú jej skokovú zmenu. Vďaka navrhnutému predikčnemu algoritmu riadenia a dobrým modelovacím algoritmom je prerégulovanie časového priebehu výstupnej veličiny takmer nulové.

Prediktívne GPC riadenie s Fuzzy Hammersteinovým modelom je vhodné aj pre menej presné modely. Problém môže spôsobiť vhodnosť procesu. GPC riadenie nedokáže riadiť systémy s nestabilnou nulou. Ak je systém vhodný pre riadenie pomocou GPC, je výhodnejšie použiť pri riadení FH model, aj keď je nepresnejší, ako model s neurónovou sieťou. Fuzzy Hammersteinov model je vhodný najmä na modelovanie nelineárnych procesov so statickou nelinearitou

PodĎakovanie: Tento článok vznikol s podporou Vega grantu 0822/08 a Vega grantu 1/3100/06.

Literatúra

- [1] Kozák, Š.: Advanced Control Methods. Bratislava: FEI STU, september 2002, 280 s. Interné materiály
- [2] Spišiak, M.: Artificial Neural Network Optimal Structures Determination. Bratislava: FEI STU, 2006. 90 s. Dizertačná práca
- [3] Doyle III., F. J., Pearson, R. K., Ogunnaike B. A.: Identification and Control Using Volterra Models. London: Springer 2002, 314 s.
- [4] Abonyi, J. et al.: Identification and Control of Nonlinear Systems Using Fuzzy Hammerstein Models. Bratislava: FEI STU, 30 s. Interné materiály
- [5] Camacho, E. F., Bordons, C.: Model Predictive Control. London: Springer – Verlag 2004, 405 s., s. 282 – 288

Kontakt:

stefan.kozak@stuba.sk