

UMĚLÉ NEURONOVÉ SÍTĚ – ZÁKLADY TEORIE A APLIKACE (17)

TAUFER I., DRÁBEK O., DOLEŽEL P.

Univerzita Pardubice, Fakulta elektrotechniky a informatiky, Katedra řízení procesů, ivan.taufer@upce.cz

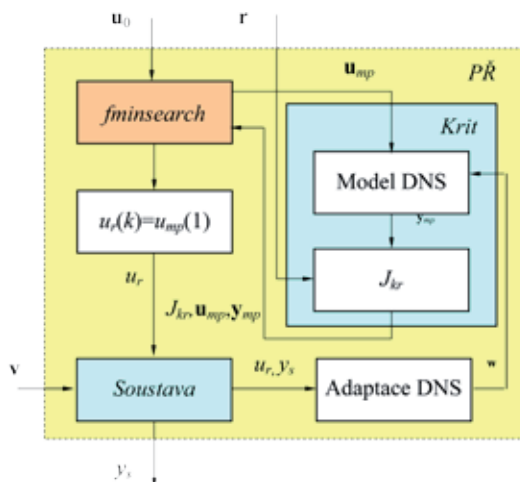
Príspevek je věnován popisu algoritmu prediktivního řízení (PŘ). Algoritmus bloku PŘ a algoritmy vybraných funkcí jsou prezentovány formou vývojových diagramů. Na jejich základě byl pak sestaven v programovém prostředí MATLAB příslušný výpočetní program, umožňující simulaci různých režimů PŘ.

17 Programová realizace simulace prediktivního řízení

17.1 Algoritmus prediktivního řízení (PŘ)

Na základě blokového schématu a popisu funkce PŘ, které jsou uvedeny v [3] na obr. 16.5, byl pro simulaci prediktivního řízení sestaven ve výpočetním systému MATLAB simulační program. Programové schéma bloku PŘ je na obr. 17.1. Blok využívá systémovou funkci výpočetního systému MATLAB pro optimalizaci – *fminsearch* (metoda simplexů Nelder a Meada) a dvě uživatelské funkce: funkci pro výpočet hodnoty kriteriální funkce včetně modelu soustavy ve tvaru DNS a funkci řízení soustavy. Součinnost mezi jednotlivými funkcemi lze popsat ve stručnosti následovně: v každém kroku řízení k je volána funkce *fminsearch*. Tato funkce pak v každém kroku optimalizace volá uživatelskou funkci *Krit*, ve které je s využitím modelu soustavy ve tvaru DNS počítána hodnota kriteriální funkce J_{kr} , (viz rovnici (16.14) v [3]). Po ukončení procedury optimalizace obdržíme vektor optimálních predikovaných hodnot akční veličiny \mathbf{u}_{mp} a odpovídající vektor predikovaných hodnot regulované veličiny \mathbf{y}_{mp} . Prvního prvku $u_{mp}(1)$ vektoru predikovaných akčních veličin \mathbf{u}_{mp} je využito jako optimálního akčního signálu $u_r(k)$, přivedeného na vstup regulované soustavy. Odpovídající výstupní hodnota regulované veličiny je $y_s(k)$. Vstupem soustavy jsou i poruchy v a akční i regulované veličině v . Vektor vstupně/výstupních hodnot regulované soustavy následně slouží i pro on-line adaptaci dopředné neuronové sítě (aktualizaci hodnot vah spojení \mathbf{w}). Činnost výše uvedeného bloku a funkcí názorně vysvětlíme s použitím vývojových diagramů. Pro zjednodušení výkladu budou použity horizonty predikce $N_1 = 1$ a $N_2 = 2$ (rovnice (16.14a)) a horizont řízení $N_u = 1$ (rovnice (16.14b)), [3]. Budeme tedy pracovat s $n = N_2 - N_1 + 1 = 2$ nezávisle proměnnými a se simplexem o $n + 1 = 3$ vrcholech.

Obr. 17.1 – Programové schéma bloku PŘ

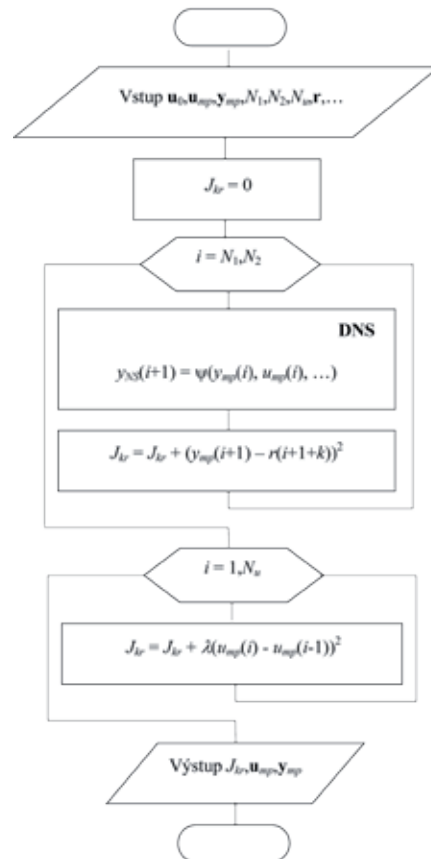


17.1.1 Funkce pro výpočet hodnoty kriteriální funkce

Funkce pro výpočet hodnoty kriteriální funkce představuje ústřední člen prediktivního řízení. Je naprogramován jako uživatelská funkce. Vývojový diagram uživatelské funkce určené k výpočtu hodnoty kriteriální funkce J_{kr} (16.14) je uveden na obr. 17.2. Vstupem do této

funkce jsou proměnné, potřebné pro kompletní výpočet hodnoty kriteriální funkce: startovací vektor akční veličiny \mathbf{u}_0 , vektor predikovaných hodnot akční veličiny \mathbf{u}_{mp} a vektor predikovaných hodnot regulované veličiny \mathbf{y}_{mp} , vypočtené v předcházejících krocích řízení, vektor žádaných hodnot \mathbf{r} , hodnoty horizontu predikce a řízení N_1 , N_2 a N_u a parametry modelu soustavy ve tvaru dopředné neuronové sítě (DNS), která pracuje ve funkci prediktoru.

Obr. 17.2 – Vývojový diagram algoritmu výpočtu hodnoty účelové funkce



Výpočet hodnoty této kriteriální funkce je realizován ve dvou sumacích cyklech.

V prvním cyklu, v rozsahu mezních hodnot parametru cyklu i od $N_1 = 1$ do $N_2 = 2$, se počítá součet čtverců odchylek (16.14a) mezi žádanou (referenční) hodnotou a predikovanou hodnotou regulované veličiny, která je výstupem dopředné neuronové sítě. DNS je před spuštěním celého řídicího algoritmu natrénována off-line a v průběhu řízení jsou pak v režimu on-line její parametry upřesňovány (režim průběžné adaptace).

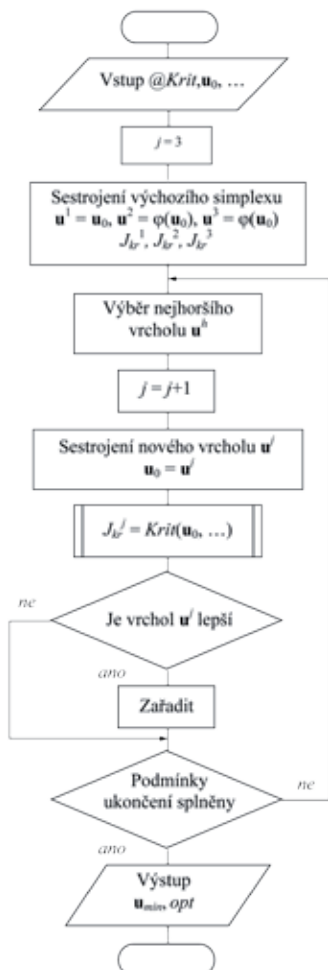
V druhém cyklu, v rozsahu mezních hodnot parametru cyklu i od 1 do $N_u = 1$, se počítá hodnota penalizačního členu (16.14b), která se přičítá k hodnotě vypočtené z rovnice (16.14a).

Výstupem popsané procedury jsou mj. hodnota kriteriální funkce J_{kr} , vektor predikovaných hodnot akční veličiny \mathbf{u}_{mp} a vektor predikovaných hodnot regulované veličiny \mathbf{y}_{mp} .

17.1.2 Metoda simplexů Nelder a Meada

Pro minimalizaci kritériální funkce (16.14) [3] byla použita systémová funkce *fminsearch*, optimalizační metoda simplexů Nelder a Meada výpočetního prostředí MATLAB. Zjednodušený vývojový digram tohoto algoritmu je uveden na obr. 17.3. S jeho podrobným vývojovým diagramem se čtenář může seznámit v [4] a s metodou simplexů mj. v [5] – [8].

Obr. 17.3 – Zjednodušený vývojový diagram algoritmu Nelder a Meada *fminsearch*



Jak bylo uvedeno výše, pracujeme s $n = 2$ nezávisle proměnnými, z čehož plyne, že simplex má $n + 1 = 3$ vrcholy.

Vstupem systémové funkce *fminsearch* je mj. symbolické jméno funkce *@Krit*, jejíž minimální hodnota je hledána, a startovací vektor predikovaných hodnot akční veličiny \mathbf{u}_0 , představující první vrchol simplexu. Dále jsou vstupními parametry všechny veličiny potřebné pro další výpočty v této funkci.

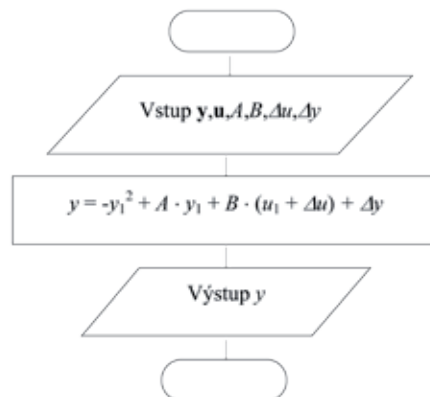
Po aktivaci funkce *fminsearch* je sestaven výchozí simplex ve tvaru trojúhelníku a v jeho vrcholech jsou pomocí uživatelské funkce *Krit* vypočítány hodnoty kritériální funkce J_{kr} . Poté je vybrán nejhorší vrchol (vrchol s největší hodnotou J_{kr}) a sestrojen nový, v pořadí $j = j + 1$ vrchol \mathbf{u}^j . Následuje volání funkce *Krit* a počítá se odpovídající hodnota J_{kr}^j . Pokud je hodnota lepší (menší) než původní J_{kr}^{j-1} , je vrchol \mathbf{u}^j zařazen do simplexu a jsou testovány podmínky pro ukončení hledání minima. Pokud je alespoň jedna z definovaných podmínek ukončení splněna, výpočet je ukončen. Pokud ne, výpočet se vrací a dochází k vyhledání dalšího, v pořadí nejhoršího vrcholu.

Výstupem funkce *fminsearch* je vektor optimálních predikovaných hodnot akční veličiny \mathbf{u}_{min} , zajišťující v daném kroku řízení k minimální hodnotu kritériální funkce J_{kr} . Její hodnota je uložena v proměnné *opt*.

17.1.3 Algoritmus uživatelské funkce pro výpočet odezvy soustavy na vstupní signál

Jednoduchý algoritmus, který řeší dále uvedenou diferenční rovnici regulované soustavy (17.1). Rovnice je doplněna o možnost zavádění poruch v akční a regulované veličině Δu a Δy . Vývojový diagram zmíněného algoritmu je uveden na obr. 17.4. Vstupem jsou předcházející hodnoty akční a regulované veličiny u a y , parametry rovnice A a B a hodnoty poruch v akční a v regulované veličině Δu a Δy . Výstupem je nová hodnota regulované veličiny.

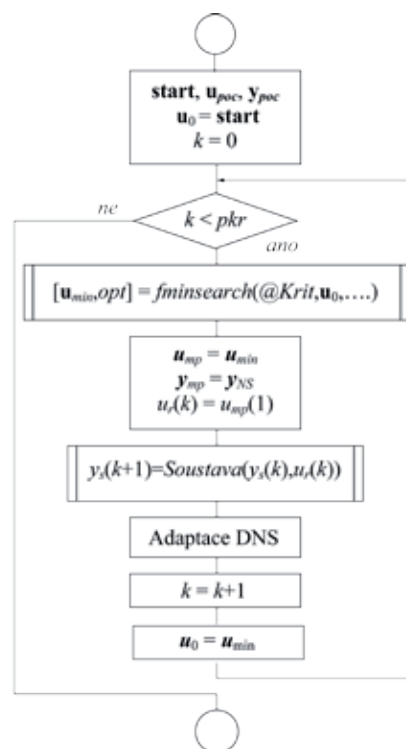
Obr. 17.4 – Vývojový diagram algoritmu soustavy



17.1.4 Algoritmus bloku prediktivního řízení

Vývojový diagram algoritmu prediktivního řízení jakožto celku, které je popsáno např. v [9] – [13], je uveden na obr. 17.5.

Obr. 17.5 – Vývojový diagram segmentu algoritmu prediktivního řízení



V úvodu bloku PŘ jsou definovány startovací hodnoty predikované akční veličiny *start* a počáteční podmínky řešení diferenční rovnice soustavy (17.1) \mathbf{u}_{poc} a \mathbf{y}_{poc} . V cyklickém režimu, který je omezen počtem kroků řízení *pkr*, je nejprve volána systémová funkce *fminsearch*. Vypočítané optimální hodnoty vektoru akční veličiny \mathbf{u}_{min} a regulované veličiny \mathbf{y}_{NS} jsou uloženy do vektorů predikovaných veličin \mathbf{u}_{mp} a \mathbf{y}_{mp} . Hodnota prvního prvku vektoru predikovaných akčních veličin $u_{mp}(1)$ je pak použita jako akční veličina v daném kroku řízení $u_s(k)$ a přivedena na vstup řízené soustavy. Je volána uživatelská funkce pro výpočet hodnoty výstupní veličiny regulované soustavy, která je uložena do proměnné $y_s(k+1)$. Hodnoty vstupních a výstupních veličin soustavy jsou zároveň použity v následném

Dokončení na další straně

bloku adaptace DNS. Adaptace proběhne klasickým trénováním DNS v režimu on-line [14] a [15].

17.2 Příklad

17.2.1 Objekt řízení

Předpokládejme, že chování řízené soustavy lze popsat nelineární diferenční rovnicí bez dopravního zpoždění [1]

$$y_s(k) + y_s^2(k-1) - A y_s(k-1) = B u(k-1), \quad (17.1)$$

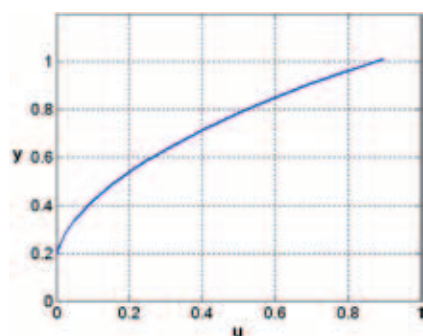
kde $u(k)$ je vstupní signál,

$y_s(k)$ – výstupní signál regulované soustavy.

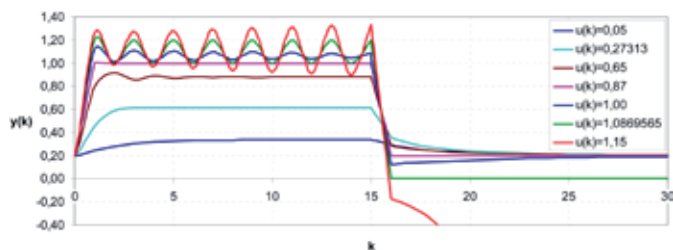
Jmenovité hodnoty parametrů jsou $A = 1,2$ a $B = 0,92$.

Detailní analýzu specifikované soustavy čtenář nalezne v [2]. Jedná se o soustavu nelineární v ustálených stavech i v dynamických vlastnostech a její pracovní rozsah je v rozmezí vstupní veličiny $u \in (0; 1)$ a výstupní veličiny $y_s \in (0,2; 1)$, viz obr. 17.6 a obr. 17.7.

Obr. 17.6 – Statická charakteristika soustavy



Obr. 17.7 – Odezvy soustavy na impulsy o různé velikosti s délkou trvání $\tau = 15T$



17.2.2 Průběh prediktivního řízení

Nechť průběh žádané hodnoty regulované veličiny (skoková změna v 5. kroku řízení) je dán vztahem

$$r(k) = \begin{cases} 0,2 & \text{pro } 0 \leq k < 5 \\ 0,6 & \text{pro } 5 \leq k \leq 10 \end{cases} \quad (17.2)$$

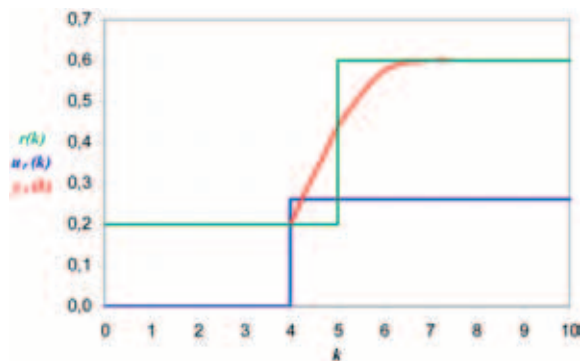
Jak bylo uvedeno výše, pro možnost názorné demonstrace volíme horizonty predikce $N_1 = 1$ a $N_2 = 2$ (v rovnici (16.14a)) a horizont řízení $N_u = 1$ (v rovnici (16.14b)). Dále byl zvolen koeficient penalizace $\lambda = 0,55$ (v rovnici (16.14b)) a vektor startovacích hodnot predikované akční veličiny $\mathbf{u}_0 = [0,4; 0,4]$. Počáteční hodnoty řešení (počáteční podmínky) dané diferenční rovnicí (17.1) jsou $u_{poc} = 0$ a $y_{poc} = 0,2$.

Vypočtené průběhy veličin v 5. kroku prediktivního řízení jsou uvedeny na obr. 17.8. Akční zásah, podle očekávání pro $N_u = 1$, předbíhá změnu žádané hodnoty o jeden krok řízení. K ustálení regulačního pochodu dochází prakticky po třech krocích řízení po změně žádané hodnoty. Jak bude dále ukázáno, okamžik ustálení lze v určitém rozmezí měnit změnou hodnoty koeficientu penalizace λ .

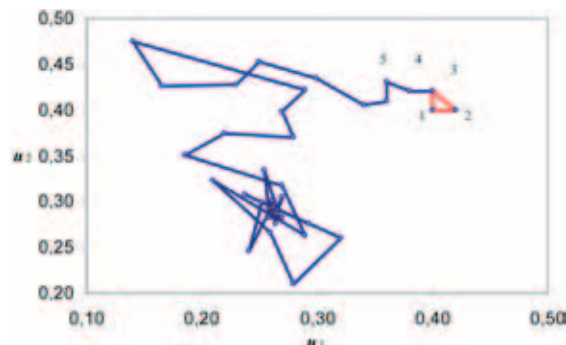
17.2.3 Analýza jednoho kroku řízení

Z výše uvedeného plyne, že základem prediktivního řízení je nalezení optimálních predikovaných hodnot akční veličiny s využitím modelu řízené soustavy, prezentované dopřednou neuronovou sítí, které zajišťují minimum kritériální funkce J_{kr} . Průběh tohoto hledání

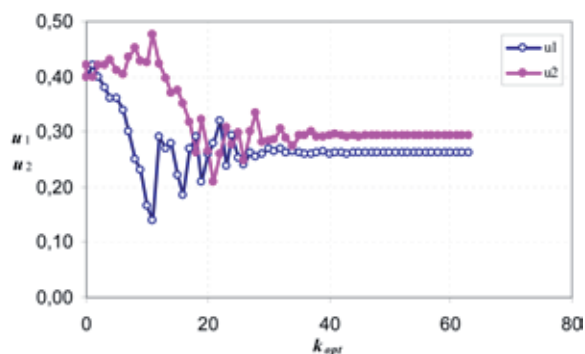
Obr. 17.8 – Průběh prediktivního řízení



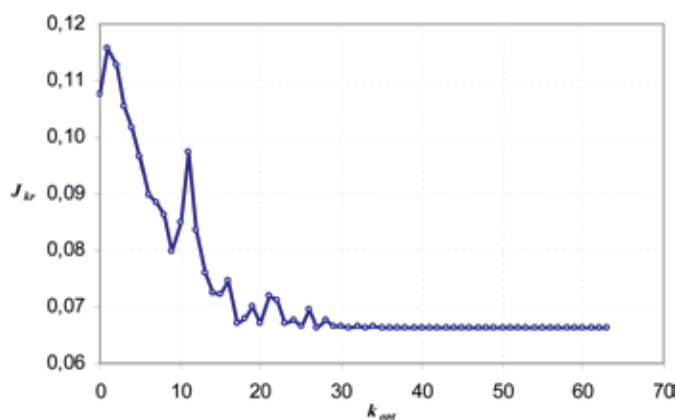
Obr. 17.9 – Sled souřadnic vrcholů simplexu při postupu k minimu kritériální funkce



Obr. 17.10 – Průběhy predikovaných hodnot akční veličiny



Obr. 17.11 – Průběh kritériální funkce



s využitím systémové funkce *fminsearch* (metody simplexů Nelder a Meada) je uveden na obr. 17.9. Startovací vektor $\mathbf{u}_0 = [0,4; 0,4]$ tvoří první vrchol simplexu $\mathbf{u}^1 = \mathbf{u}_0$. Prvky tohoto vektoru představují predikované hodnoty akčních veličin posunutých vzájemně o jeden interval vzorkování. Optimálních hodnot predikované akční veličiny, které zajišťují minimum kritériální funkce $J_{kr} = 0,0663$, bylo dosaženo po 64 iteračních krocích $\mathbf{u}_{mp} = \mathbf{u}^{64} = [0,262; 0,293]$. Průběhy predikovaných hodnot akční veličiny jsou uvedeny na obr. 17.10 a průběh hodnoty kritériální funkce na obr. 17.11. První prvek

vektoru optimálních predikovaných hodnot u_{mp} se dále použije jako optimální hodnota akčního signálu prediktivního řízení v 5. kroku řízení $u_1(5) = u_{mp}(1) = 0,262$.

Problematika je řešena v rámci výzkumného záměru MŠM 0021627505 „Řízení, optimalizace a diagnostika složitých systémů“ a programu vědeckovýzkumné spolupráce ČR a SR KONTAKT MŠMT č. MEB 0810003 „Identifikace a řízení složitých nelineárních soustav s využitím metod umělé inteligence“.

Literatura

- [1] KOZÁK, Š.; KAJAN, S. Adaptive Control of Nonlinear Dynamical Systems Using Neural Networks. In *Proceeding of the 2nd Slovak Conference on Artificial Neural Networks SCANN 98*, November 10–12, 1998, pp. 72–76.
- [2] TAUFER, I.; DRÁBEK, O. Adaptive Control of Non-linear System using Neural Network. *Acta Mechanica Slovaca*. Vol. 14, No 1/2010. Košice (Slovak Republic) : TU Košice, 2010. pp. 54–63. ISSN 1335-2393.
- [3] TAUFER, I.; DRÁBEK, O. Umělé neuronové sítě – základy teorie a aplikace (16). *CHEMAGAZÍN*, 2 (XXI), 2011, s. 45–47. ISSN 1210-7409.
- [4] TAUFER, I.; DRÁBEK, O.; JAVŮREK, M. Metoda simplexů – efektivní nástroj pro řešení optimalizačních úloh. *CHEMAGAZÍN*, 6 (XX), 2010, s. 31–34. ISSN 1210-7409.
- [5] NELDER, J. A.; MEAD, R. A simplex method for function minimization. *Computer Journal*, vol. 7, 1965, p. 308.
- [6] BOX, M. J. A new method of constrained optimization and comparison with other methods. *Computer Journal*, vol. 8, 1965, No 1, p. 42–52.
- [7] SPENDLEY, W.; HEXT, G. R.; HIMSWORTH, F. R. Sequential application of simplex designs in optimization and Evolutionary Operation. *Technometrics*, vol. 4, No 4, p. 441
- [8] KRUG, G. K.; TAUFER, I. Simplexní metoda plánování experimentů. *Technika v chemii*, 1974, č.57, s. 1–8.
- [9] MIKLEŠ, J.; FIKAR, M. *Modelovanie, identifikácia a riadenie procesov 2. Identifikácia a optimálne riadenie*. Bratislava: Vydavateľstvo STU Bratislava, 2004. ISBN 80 227 2134-4.
- [10] CAMACHO, E. F.; BORDONS, C. *Model Predictive Control*. London : Springer, 405 p. ISBN 1-85233-694-3
- [11] LAZAR, M.; PASTRAVANU, O. A neural predictive controller for non-linear systems. *Mathematics and Computers in Simulation*, 60 (2002), pp. 315–324.
- [12] SKLIARENKO, E.G. Prediktívne regulovanie sľorosti elektroprivoda robotov na baze neiromodeli [on line]. *Elektromashinoborudovannia ta elektroobladnannia*. 52. [cit. 11.8.2009]. Dostupné na „http://www.nbu.gov.ua/Articles/OSPU/ee_52/2.htm“.
- [13] NN Predictive Control. [on line]. *Neural Network Toolbox*. [cit. 11.8.2009]. Dostupné na „<http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/control3.html>“.
- [14] TAUFER, I.; DRÁBEK, O. Umělé neuronové sítě – základy teorie a aplikace (8). *CHEMAGAZÍN*, 4 (XVII), 2007, s. 28–30. ISSN 1210-7409.
- [15] TAUFER, I.; DRÁBEK, O. Umělé neuronové sítě – základy teorie a aplikace (9). *CHEMAGAZÍN*, 6 (XVII), 2007, s. 32–4. ISSN 1210-7409.

Abstract:

THE ARTIFICIAL NEURAL NETWORKS – BASIC THEORY AND APPLICATION

Summary: Paper deals with predictive control algorithm. Different approaches are presented as flow charts. On their base program for different modes of the predictive control is created in computational environment MATLAB.

Key words: Model predictive control, neural network, simplex method optimization