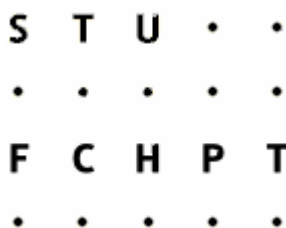


# **SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**

Fakulta chemickej a potravinárskej technológie

**Ústav informatizácie, automatizácie a matematiky**

**Oddelenie informatizácie a riadenia procesov**



## **DIPLOMOVÁ PRÁCA**

### **Prediktívne riadenie s fuzzy modelom**

Vypracovala:

**Bc. Jolana Vlkanová**

Vedúca diplomovej práce:

**Ing. Anna Vasičkaninová**

**B r a t i s l a v a 2007**

***Prehlásenie:***

Čestne prehlasujem, že som diplomovú prácu vypracovala samostatne podľa pokynov vedúceho diplomovej práce a s použitím literatúry, ktorú uvádzam v osobitnom zozname. Ďalej prehlasujem, že súhlasím s akýmkoľvek prípadným využitím diplomovej práce ústavom.

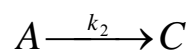
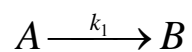
*V Bratislave, 20. mája 2007*

.....

*Na tomto mieste by som sa chcela poďakovať Ing. Anne Vasičkaninovej za odborné vedenie, cenné rady a pripomienky i všetkým ostatným ľuďom, ktorí mi akoukoľvek formou pomohli pri vypracovaní mojej diplomovej práce.*

## Abstrakt

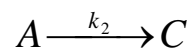
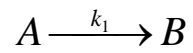
Práca sa zaoberá prediktívnym riadením s fuzzy modelom. Predmetom riadenia je prietokový chemický reaktor. V reaktore sa odohrávajú dve exotermické paralelné reakcie:



Nelineárny model reaktora použijeme na získanie údajov pre návrh fuzzy modelov. Treba navrhnuť taký fuzzy model, ktorý opisuje správanie sa chemického reaktora. Stratégia riadenia je založená na kombinácii prediktívneho regulátora a fuzzy modelu.

## Abstract

This work deals with predictive control with fuzzy model. The object of control is continued stir tank reactor. Two parallel exothermic reactions take place in the reactor:



Non-linear model of reactor will be used to obtain data for designing of fuzzy models. It is necessary to find fuzzy model, that describes behaving of continued stirred tank reactor the most precisely. Strategy of controlling is based on a combination of predictive controller and fuzzy model.

## Obsah

Úvod .....	9
1. Prediktívne metódy riadenia .....	10
1.1 Princíp prediktívneho riadenia .....	10
1.2 Model riadenia .....	12
1.3 Neurónové prediktívne riadenie .....	12
1.4 Účelová funkcia .....	13
1.5 Optimalizácia účelovej funkcie .....	14
1.6 Štruktúra neurónovej siete .....	17
1.7 Predikcia s použitím neurónovej siete .....	18
1.8 Návrh prediktívneho regulátora pomocou Matlab .....	20
1.8.1 Identifikácia systému .....	21
1.8.2 Prediktívne riadenie .....	22
2. Fuzzy logika .....	23
2.1 Úvod do fuzzy logiky .....	23
2.2 Fuzzy množiny .....	24
2.3 Operácie s fuzzy množinami .....	25
2.4 Funkcia príslušnosti .....	27
2.5 Lingvistická premenná .....	28
2.6 Inferenčné pravidlá .....	29
2.7 Fuzzy model .....	29
2.8 Fuzzy inferenčné systémy .....	30
2.9 Tvorba fuzzy modelov metódou Anfi .....	32
3. Prietokový chemický reaktor .....	34
3.1 Popis chemického reaktora .....	34
3.2 Dynamický matematický model reaktora .....	36

---

3.3	Matematický model rovnovážného stavu a analýza stability .....	38
3.4	Analýza prietoku chladiaceho média .....	40
4.	Vytvorenie fuzzy modelov .....	42
4.1	Návrh fuzzy modelu Takagi – Sugenovho typu .....	42
4.1.1	Inicializácia pomocou príkazu genfis1 .....	43
4.1.2	Inicializácia pomocou príkazu genfis2 .....	47
4.2	Návrh fuzzy modelu pre viac vstupných premenných .....	50
4.3	Fuzzy model s náhodným signálom na vstupe .....	54
5.	Návrh prediktívneho riadenia .....	58
5.1	Nastavenie prediktívneho regulátora .....	58
5.2	Identifikácia procesu .....	59
5.3	Prediktívne riadenie chemického reaktora .....	64
	Záver .....	65
	Literatúra .....	66
	Príloha .....	68
A.	Program na návrh fuzzy modelov fsgenfis1.fis a fsgenfis2.fis .....	68
B.	Program pre návrh fuzzy modelu pre viac vstupných premenných .....	69
C.	Štruktúra fuzzy inferenčného systému fis1.fis .....	70
D.	Štruktúra fuzzy inferenčného systému fis2.fis .....	71

## Úvod

Prediktívne metódy riadenia predstavujú v súčasnosti veľkú skupinu moderných metód riadenia s čoraz väčším počtom aplikácií. Tieto regulátory sa často krát používajú pri riadení procesov, ktoré sú neminimálne fázové, nestabilné a pre mnohorozmerné procesy s dopravným oneskorením.

Pod pojmom prediktívne riadenie chápeme triedu metód riadenia, ktoré matematický model systému používajú na predikciu budúceho riadeného výstupu systému. Výpočet postupnosti budúcich riadiacich zásahov zahŕňa minimalizáciu vhodnej účelovej funkcie s budúcimi trajektóriami prírastku riadenia a regulačnej odchýlky.

Jednou z možností ako odvodiť fuzzy model je metóda *Adaptive Neuro Fuzzy Inference System* (ANFIS metóda). Podstatou tejto metódy je neuro – adaptívny proces učenia, na základe ktorého možno z nameraných údajov odvodiť parametre funkcií príslušnosti a extrahovať bázu pravidiel, ktoré použijeme na návrh fuzzy modelu.

Cieľom tejto práce je navrhnúť prediktívne riadenie s fuzzy modelom. Predmetom riadenia je prietokový chemický reaktor s dvoma exotermickými paralelnými reakciami. Nelineárny model reaktora je použitý na získanie údajov pre návrh fuzzy modelu, na ktorý je aplikovaný príkaz *anfís*. Na inicializáciu fuzzy systému sú použité príkazy *genfis1* a *genifs2*. Je potrebné navrhnúť taký fuzzy model, s jedným a dvoma vstupmi, ktorý opisuje správanie sa chemického reaktora. Pri prediktívnom riadení je možné uvažovať obmedzenia na vstupné a výstupné veličiny priamo pri návrhu regulátora, ktoré sú výsledkom analýzy chemického reaktora. Prediktívne riadenie pomocou fuzzy modelov bude porovnané s prediktívnym riadením s modelom reaktora v tvare s-funkcie, ktorá zároveň predstavuje riadený proces.

# 1 Prediktívne metódy riadenia

Prediktívne metódy riadenia predstavujú v súčasnosti veľkú skupinu moderných metód riadenia s čoraz väčším počtom aplikácií.

V roku 1987 Clarke so svojimi spoločníkmi prvýkrát predstavil prediktívne riadenie, avšak už v roku 1970 ho prvýkrát úspešne použili pri riadení chemického procesu. Po tomto období sa prediktívne regulátory častejšie používali pri riadení procesov, pretože sú aplikovateľné na neminimálne fázové, nestabilné a pre mnoho - rozmerové procesy s dopravným oneskorením [14].

Pod pojmom prediktívne riadenie rozumieme triedu metód riadenia, ktoré spájajú určité spoločné črty :

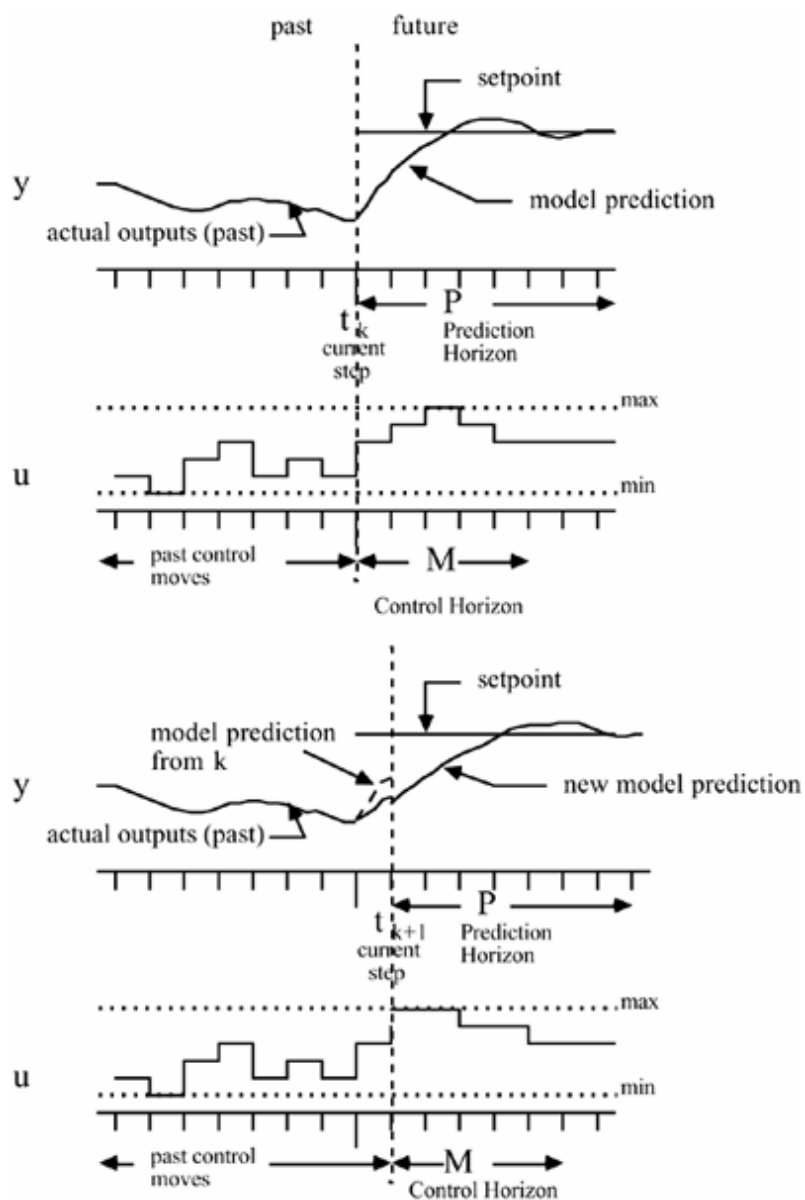
- matematický model riadenia systému je použitý na predikciu budúceho riadeného výstupu systému,
- je známa trajektória žiadanej veličiny do budúcnosti,
- výpočet postupnosti budúcich riadiacich zásahov zahŕňa minimalizáciu vhodnej účelovej funkcie s budúcimi trajektóriami prírastku riadenia a regulačnej odchýlky,
- iba prvý akčný zásah je realizovaný a celý postup minimalizácie funkcionálu sa opakuje v ďalšej perióde vzorkovania.

Algoritmy prediktívneho riadenia sú pri riadení procesov mnohostranne použiteľné a robustné. Kvalita riadenia je obvykle vyššia ako u PID regulátorov [9].

## 1.1 Princíp prediktívneho riadenia

Model procesu je použitý na predikciu budúcich výstupov procesu. V každom kroku  $k$  je potrebné vyriešiť problém optimalizácie (obr.1.1) [5]. Predikcie sú vypočítané vzhľadom k informáciám dostupným do času  $k$  a tiež vzhľadom k neznámej trajektórii akčných zásahov, ktoré treba určiť. Trajektória budúcich akčných zásahov je určená z riešenia optimalizačného problému, obsahujúceho vhodnú účelovú funkciu a obmedzenia. Účelová funkcia, založená na predikcii výstupov cez predikčný horizont  $P$ , je minimalizovaná akčnou veličinou, pohybujúcou sa v horizonte  $M$ . Hoci v predošlom kroku

bola počítaná celá trajektória akčných zásahov, iba prvý člen  $u(k)$  je aplikovaný do procesu. V ďalšej perióde vzorkovania je celý postup opakovaný. Tento princíp je známy ako stratégia pohyblivého horizontu [5], [9].



Obr. 1.1 Princíp prediktívneho riadenia

## 1.2 Model riadenia

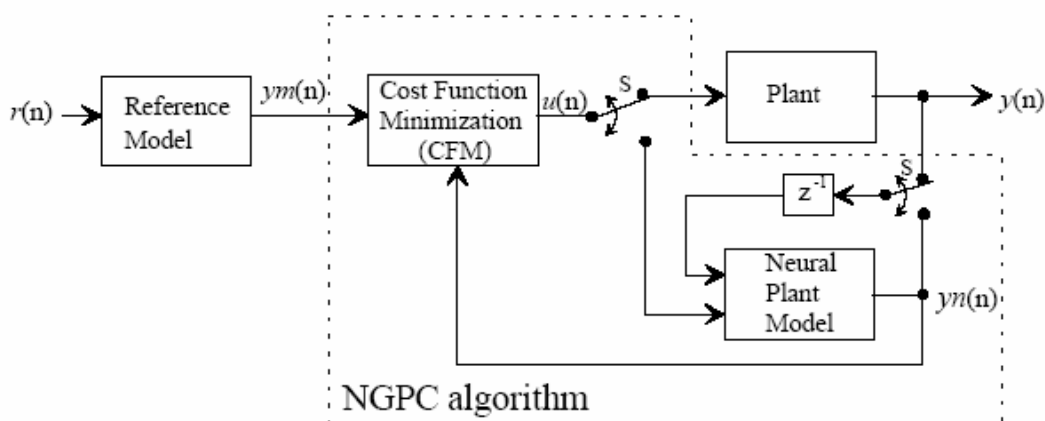
Pri prediktívnom riadení je pre výpočet predikcie budúcej výstupnej trajektórie potrebný model procesu.

Jedna z výhod prediktívneho riadenia je možnosť použiť ľubovoľný model procesu. Používaný model musí dostatočne vystihnúť dynamické vlastnosti procesu. Ako model môže byť použitá impulzná funkcia, prechodová funkcia, prenosová funkcia či stavový opis.

V súčasnosti sa často používajú aj iné typy modelov, napríklad neurónové siete či fuzzy opisy. V tomto prípade je model alebo používaný priamo, alebo sú ním iba generované niektoré charakteristiky procesu [9].

## 1.3 Neurónové prediktívne riadenia ( Neural Generalized Predictive Control )

Pri riadení nelineárneho modelu môžeme predikciu budúceho riadeného výstupu vylepšiť, ak namiesto nelineárneho dynamického modelu použijeme model navrhnutý na základe neurónovej siete [14].



Obr. 1.2 Prediktívny regulátor

Schéma sa skladá zo štyroch častí : riadený proces, referenčný model, ktorý špecifikuje požadovaný výkon procesu, neurónová sieť, ktorá modeluje proces a algoritmus minimalizácie účelovej funkcie.

Prediktívny regulátor môžeme rozdeliť na dva bloky. Prvým blokom je minimalizácia účelovej funkcií (*CFM* blok) a druhým je neurónová sieť. Systém riadenia sa začína vstupným signálom  $r(n)$ , vstupujúcim do referenčného modelu. Tento model produkuje referenčný signál  $ym(n)$  pre vstup do *CFM* bloku. *CFM* algoritmus produkuje výstup, ktorý je vstupom buď do procesu, alebo ako vstup do modelu procesu. Dvojpolohový prepínač sa prepne na proces iba vtedy, ak blok *CFM* úspešne vyriešil optimalizáciu účelovej funkcie. Do toho času je prepínač prepnutý na model procesu, pretože algoritmus *CFM* ho používa na výpočet postupnosti riadiacich zásahov  $u(n+1)$ .

Princíp prediktívneho riadenia je nasledovný [14]:

1. generovanie trajektórie žiadanej veličiny, ak trajektória žiadanej veličiny nie je známa  $ym(n)$ , najčastejšie sa uvažuje konštantná a rovná aktuálnej žiadanej veličine,
2. výpočet postupnosti predchádzajúcich riadiacich vstupných zásahov a predikcia procesu,
3. výpočet nového riadiaceho vstupu, ktorý minimalizuje účelovú funkciu,
4. opakovanie bodov 2 a 3 pokiaľ požadovaná minimalizácia nie je dosiahnutá,
5. odoslanie prvého riadiaceho vstupu do procesu,
6. opakovanie celého postupu pre každý časový okamih.

## 1.4 Účelová funkcia

Štandardná účelová funkcia obsahuje kvadratické členy regulačnej odchýlky a prírastkov riadenia na konečnom horizonte do budúcnosti [14],

$$J = \sum_{j=N_1}^{N_2} [ym(n+j) - yn(n+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(n+j)]^2 \quad (1.4.1)$$

kde :

$y_m$  je trajektória žiadanej veličiny,

$y_n$  je výstup  $j$  krokov, predikovaný v budúcnosti vzhľadom k informáciám dostupným do času  $k$ ,

$\Delta u$  je postuponosť budúcich prírastkov riadenia, ktoré majú byť vypočítané,

$N_1$  je minimálny horizont ,

$N_2$  je maximálny horizont ,

$N_u$  je riadiaci horizont ,

$\lambda$  je penalizačný faktor riadenia.

$N_1$  a  $N_2$  určujú interval v budúcnosti, kedy je potrebné sledovať trajektóriu žiadanej veličiny. Hodnoty  $N_1$  a  $N_u$  je potrebné zvoliť tak, aby ich hranica bola menšia alebo rovná  $N_2$ .  $N_1$  sa volí minimálne  $T_{d+1}$ , kde  $T_d$  je približná hodnota dopravného oneskorenia procesu. Týmto spôsobom sa môžeme vyhnúť problémom s neminimálne fázovými procesmi. Hodnota  $N_2$  by mala pokrývať dôležitú časť prechodovej charakteristiky. Obyčajne sa volí porovnateľná s časom  $T_{90}$ , čo je čas, za ktorý výstup prejde z 10% na 90% svojej ustálenej hodnoty.

Druhá strana účelovej funkcie obsahuje aj penalizačný faktor riadenia, ktorý je potrebný na kontrolovanie rovnováhy medzi dvoma súčtami. Faktor  $\lambda$  vystupuje ako tlmič na predikciu  $u(n+1)$ . Využitie riadiaceho horizontu  $N_u$  znižuje výpočtovú náročnosť metódy [9], [14].

## 1.5 Optimalizácia účelovej funkcie

Prediktívne riadenie je založené na optimalizácii účelovej funkcie v bloku *CFM*. V tomto bloku sa minimalizuje účelová funkcia vzhľadom na  $[u(n+1), u(n+2), \dots, u(n+N_u)]^T$ , ktorú označujeme  $U$ . Táto problematika je riešená nastavením Jacobiana účelovej funkcie (1.5.1) na nulu. Ako *CFM* algoritmus sa používa Newton - Raphsonovu metóda, pomocou ktorej je  $J$  iteračne minimalizovaná funkcia, aby sa určilo čo najlepšie  $U$ . Pre každú iteráciu z  $J(k)$  je vygenerovaný aj riadiaci vstup  $U(k)$ :

$$U(k) = \begin{bmatrix} u(n+1) \\ u(n+2) \\ \vdots \\ u(n+N_u) \end{bmatrix} \quad (1.5.1)$$

Pre  $U(k+1)$  má Newton-Raphson pravidlo tvar :

$$U(k+1) = U(k) - \left( \frac{\partial^2 J}{\partial U^2}(k) \right)^{-1} \frac{\partial J}{\partial U}(k) \quad (1.5.2)$$

kde Jacobian je:

$$\frac{\partial J}{\partial U}(k) = \begin{bmatrix} \frac{\partial J}{\partial u(n+1)} \\ \vdots \\ \frac{\partial J}{\partial u(n+N_u)} \end{bmatrix} \quad (1.5.3)$$

a Hessian predstavuje :

$$\frac{\partial^2 J}{\partial U^2}(k) = \begin{bmatrix} \frac{\partial^2 J}{\partial u(n+1)^2} & \cdots & \frac{\partial^2 J}{\partial u(n+1)\partial u(n+N_u)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial u(n+N_u)\partial u(n+1)} & \cdots & \frac{\partial^2 J}{\partial u(n+N_u)^2} \end{bmatrix} \quad (1.5.4)$$

Priame vyriešenie rovnice (1.5.2) vyžaduje inverznú Hessovu maticu. Tento výpočet je veľmi náročný, preto rovnicu (1.5.2) je potrebné prepísať do tvaru lineárnych rovníc  $Ax = b$ . Výsledok je nasledovný :

$$\frac{\partial^2 J}{\partial U^2}(k)(U(k+1) - U(k)) = -\frac{\partial J}{\partial U}(k) \quad (1.5.5)$$

Z rovnice (1.5.5) sa dajú vyjadriť členy  $A$ ,  $b$  a  $x$ :

$$\frac{\partial^2 J}{\partial U^2}(k) = A \quad (1.5.6)$$

$$-\frac{\partial J}{\partial U}(k) = b \quad (1.5.7)$$

$$U(k+1) - U(k) = x \quad (1.5.8)$$

Rovnicu (1.5.2) v tejto forme je možné veľmi jednoducho vyriešiť ako sústavu lineárnych rovníc. Po vypočítaní hodnoty  $x$  je ďalším krokom rovnice (1.5.8) výpočet  $U(k+1)$ . Tento postup je opakovaný, pokiaľ rozdiel medzi jednotlivými hodnotami  $U(k+1)$  nie je menší ako hodnota  $\varepsilon$ . Pri riešení  $x$  je potrebné pre každú Newton – Rhapsion iteráciu vypočítať prvky Jacobiana a Hessiana. V rovnici (1.5.9) je znázornený  $h$ -tý prvok Jacobiana.

$$\begin{aligned} \frac{\partial J}{\partial u(n+h)} = & -2 \sum_{j=N_1}^{N_2} [ym(n+j) - yn(n+j)] \frac{\partial yn(n+j)}{\partial u(n+h)} \\ & + 2 \sum_{j=1}^{N_u} \lambda(j) [\Delta u(n+j)] \frac{\partial \Delta u(n+j)}{\partial u(n+h)} \end{aligned} \quad (1.5.9)$$

Člen  $\frac{\partial \Delta u(n+j)}{\partial u(n+h)}$  môžeme rozšíriť pomocou Kroneckerových delta funkcií

$$\frac{\partial u(n+j)}{\partial u(n+h)} - \frac{\partial u(n-1+j)}{\partial u(n+h)} = \delta(h, j) - \delta(h, j-1) \quad (1.5.10)$$

V rovnici (1.5.11) je znázornený Hessian pre  $m$  a  $h$  prvok.

$$\begin{aligned} \frac{\partial^2 J}{\partial u(n+m) \partial u(n+h)} = & 2 \sum_{j=N_1}^{N_2} \left\{ \frac{\partial yn(n+j)}{\partial u(n+m)} \frac{\partial yn(n+j)}{\partial u(n+h)} - \frac{\partial^2 yn(n+j)}{\partial u(n+m) \partial u(n+h)} [ym(n+j) - yn(n+j)] \right\} \\ & + 2 \sum_{j=1}^{N_u} \lambda(j) \left\{ \frac{\partial \Delta u(n+j)}{\partial u(n+m)} \frac{\partial \Delta u(n+j)}{\partial u(n+h)} + \Delta u(n+j) \frac{\partial^2 \Delta u(n+j)}{\partial u(n+m) \partial u(n+h)} \right\} \end{aligned} \quad (1.5.11)$$

Kde  $h = 1, \dots, N_u$  a  $m = 1, \dots, N_u$

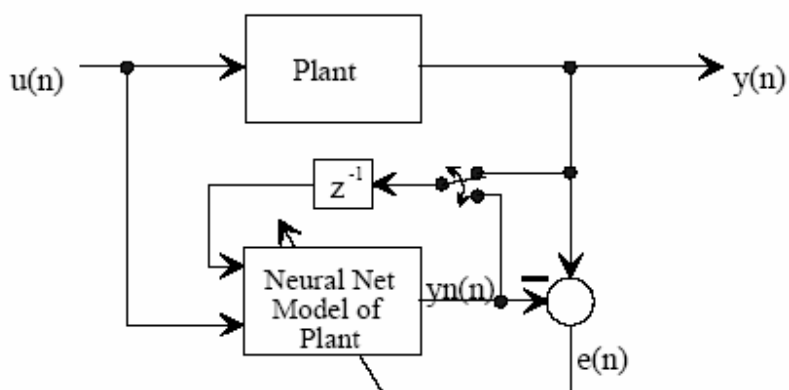
V tomto prípade je možné využiť poznatky z delta funkcie, čím sa získa rozšírenie v rovnici (1.5.12).

$$\frac{\partial \Delta u(n+j)}{\partial u(n+h)} \frac{\partial \Delta u(n+j)}{\partial u(n+m)} = (\delta(h, j) - \delta(h, j-1))(\delta(m, j) - \delta(m, j-1)) \quad (1.5.12)$$

Posledný prvok rovnice (1.5.12) je na vyhodnotenie  $U(k+1)$ . Je potrebný na výpočet ďalej derivovaného výstupu  $y(n+j)$  [14].

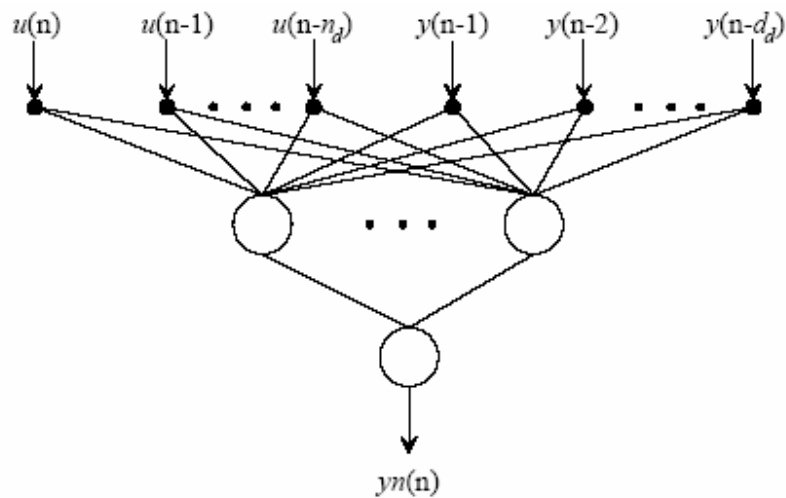
## 1.6 Štruktúra neurónovej siete

V bloku *NGPC* je modelom procesu neurónová sieť. Bloková schéma tréningu neurónovej siete na modelovanie procesu je znázornená na obr.1.3 [14]. Neurónová sieť a proces majú rovnaký vstup  $u(n)$ , okrem tohto vstupu má neurónová sieť aj ďalší vstup, ktorým môže byť výstup z procesu  $y(n)$  alebo výstup z neurónovej siete  $y_n(n)$ . Zvolenie vstupu závisí od procesu a aplikácií. Tento vstup pomáha sieti so stabilizáciou nestabilných systémov. Tréning siete je nastavený tak, aby súbor vstupov produkoval požadovaný súbor výstupov. Je vytvorená odchýlka medzi odozvami siete  $y_n(n)$  a procesu  $y(n)$ . Táto odchýlka je používaná na aktualizáciu váhy zo siete počas tréningu. Druhým vstupom  $y_n(n)$  bude dovtedy, kým odchýlka nie je redukovaná na požadovanú úroveň [6], [14].



Obr.1.3. Bloková schéma tréningu neurónovej siete

Na obr.1.4 [14] je zobrazená mnohovrstvová dopredná neurónová sieť so štruktúrou časového oneskorenia. V tomto prípade sú vstupy do siete zostavené z externých vstupov  $u(n)$  a  $y(n-1)$  a ich zodpovedajúcich oneskorených uzlov  $u(n-1), \dots, u(n-n_d)$ , a  $y(n-2), \dots, y(n-d_d)$ . Parametre  $n_d$  a  $d_d$  predstavujú počet oneskorených uzlov. Namiesto vstupu  $y(n-1)$  by mohol byť vstupom aj  $yn(n-1)$  a jeho oneskorené hodnoty. Sieť má jednu skrytú vrstvu obsahujúcu niekoľko skrytých uzlov, ktoré používajú všeobecnú výstupnú funkciu  $f_j$ . Výstupný uzol používa lineárnu výstupnú funkciu [14].



Obr. 1.4. Mnohovrstvová dopredná neurónová sieť

## 1.7 Predikcia s použitím neurónovej siete

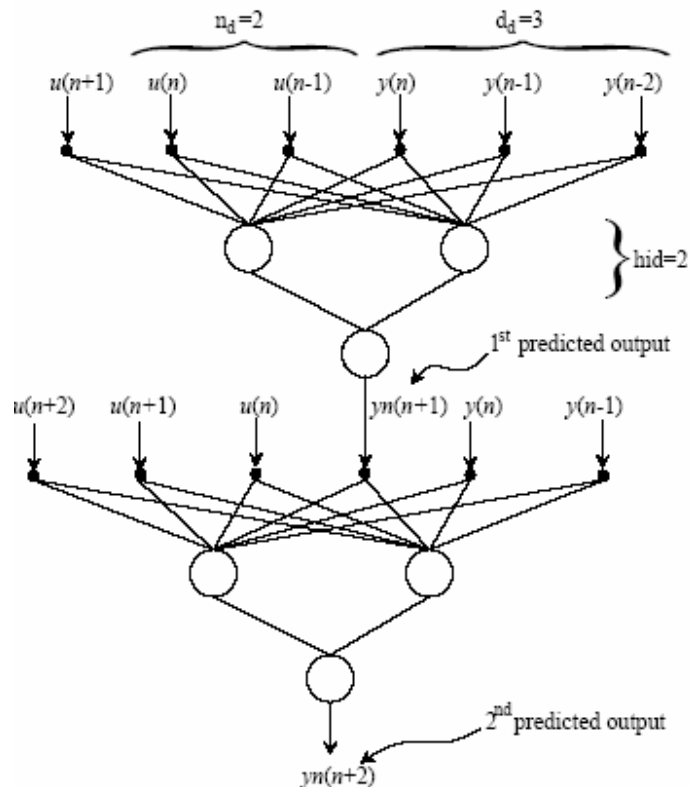
*NGPC* algoritmus používa výstup modelu procesu na predikciu dynamiky procesu k ľubovoľnému vstupu od času  $n$  do času  $n+k$ . Toto sa dosiahne použitím rovnice časového posunu a  $k$ , čím sa získajú rovnice (1.7.1) a (1.7.2). Rovnica (1.7.1) je matematickým vyjadrením obrázku z predošlej kapitoly( obr.1.4) .

$$yn(n+k) = \sum_{j=1}^{hid} \{w_j f_j(net_j(n+k))\} + b \quad (1.7.1)$$

$$\begin{aligned}
net_j(n+k) = & \sum_{i=0}^{n_d} w_{j,i+1} \begin{cases} u(n+k-i) & , k - N_u < i \\ u(n+N_u) & , k - N_u \geq i \end{cases} \\
& + \sum_{i=1}^{\min(k, d_d)} (w_{j, n_d+i+1} yn(n+k-i)) + \sum_{i=k+1}^{d_d} (w_{j, n_d+i+1} y(n+k-i)) \\
& + b_j
\end{aligned} \tag{1.7.2}$$

Zložitosť nastáva vtedy, ak sa vezmú do úvahy účelová funkcia a rekurzívna vlastnosť predikcie. Prvý súčet rozdelí vstup na dve časti reprezentované podmienkami. Podmienka  $k - N_u < i$  vedie predchádzajúce budúce hodnoty  $u$  až do  $u(n+N_u-1)$ . Podmienka  $k - N_u > i$  stavia vstupy od  $u(n+N_u)$  do  $u(n+k)$  rovným  $u(n+N_u)$ . Druhá podmienka nastane iba v prípade ak  $N_2 > N_u$ . Ďalší súčet v rovnici (1.7.2) riadi rekurzívnu časť predikcie. Toto spätne viaže výstup siete,  $yn$ , po  $k$  alebo  $d_d$  podľa toho, ktorá hodnota je menšia. Posledný člen rovnice (1.7.2) riadi predchádzajúce hodnoty  $y$ .

Nasledujúci príklad graficky reprezentuje predikciu s použitím neurónovej siete. Predpokladáme neurónovú sieť s dvoma skrytými uzlami ( $h_d = 2$ ) a s jedným výstupom. Na vstupe je  $u(n)$  a jeho dva predchádzajúce vstupy ( $n_d = 2$ ) a tri predchádzajúce výstupy ( $d_d = 3$ ). Predpokladáme dvojstupňovú predikciu, neurónová sieť potrebuje predikciu výstupu v čase  $n+1$  a v čase  $n+2$ . Tento príklad je znázornený na obr.1.5 [14]. Na obrázku vidíme predikciu procesu pre  $k = 2$ . Aby sme dostali výstup  $yn(n+2)$ , je potrebné na vstupe zaviesť  $u(n+1)$  a  $u(n+2)$ . Predikcia je odštartovaná v čase  $n$  s počiatočnými podmienkami  $[u(n) \ u(n-1)]$  a  $[y(n) \ y(n-1) \ y(n-2)]$  a odhadovaným vstupom  $u(n+1)$ . Výstup z tohto procesu je  $yn(n+1)$ , ktorý sa vracia späť do neurónovej siete na ďalšiu predikciu. Po opakovaní procesu na výstupe dostaneme  $yn(n+2)$  [14].



Obr.1.5. Predikcia

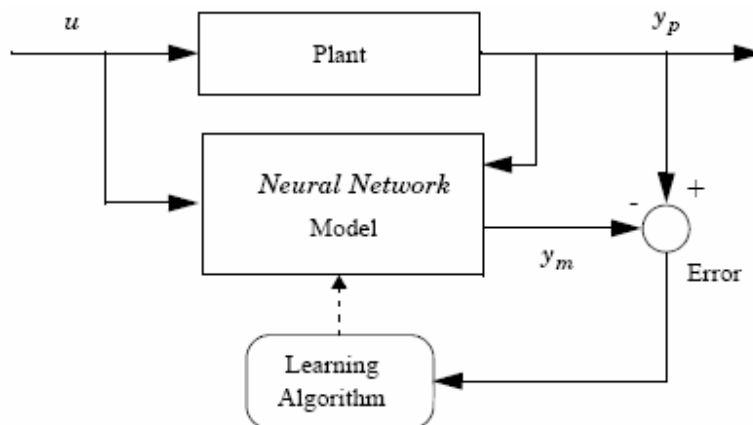
## 1.8 Návrh prediktívneho regulátora pomocou Matlab

Prediktívny regulátor na základe neurónovej siete je realizovaný v *Neural Network Toolbox*, ktorý na predikciu budúceho riadeného výstupu používa ako model neurónovú sieť a nelineárny proces [6].

Prvým krokom v modeli prediktívneho regulátora je určenie neurónovej siete, tento krok nazývame identifikáciou systému. V ďalšom kroku model procesu je použitý na predikciu budúcich výstupov procesu.

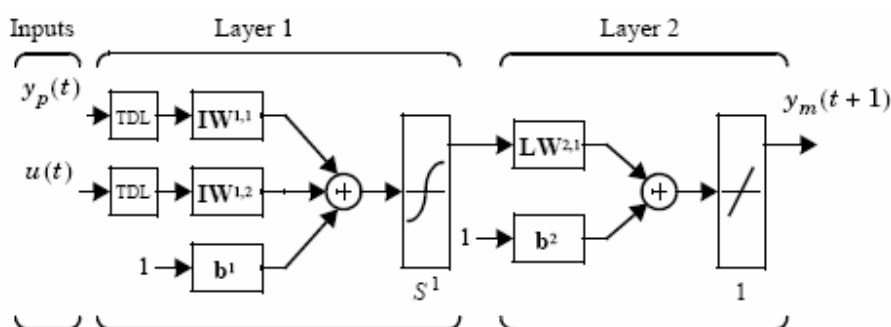
### 1.8.1 Identifikácia systému

Prvým stupňom modelovania prediktívneho regulátora je tréning neurónovej siete na určenie dynamiky procesu. Odchýlka medzi výstupom procesu a neurónovej siete je používaná na tréning signálu neurónovej siete. Tento proces reprezentuje obr.1.6 [6].



Obr.1.6. Bloková schéma tréningu v Matlab

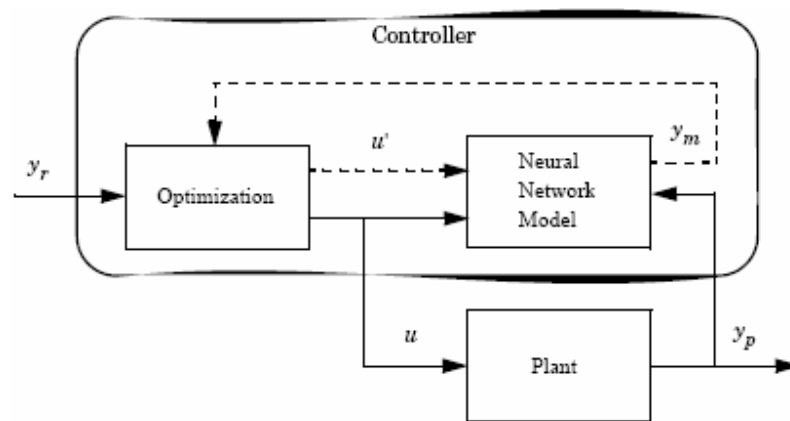
Model neurónovej siete na určenie budúcich výstupov procesu používa predchádzajúce vstupy a výstupy z procesu. Štruktúra neurónovej siete je znázornená na obr. 1.7 [6].



Obr.1.7. Štruktúra neurónovej siete pri predikcii

### 1.8.2 Pediktívne riadenie

Nasledujúca bloková schéma (obr.1.8) ilustruje proces prediktívneho riadenia. Regulátor sa skladá z modelu neurónovej siete a z optimalizačného bloku. Blok optimalizácie minimalizuje účelovú funkciu, tým určuje hodnoty  $u'$ . Pri minimalizácii dosiahnutý optimálny  $u$  je vstupom do procesu [6].



Obr. 1.8. Bloková schéma prediktívneho regulátora

## 2 Fuzzy logika

### 2.1 Úvod do fuzzy logiky

Fuzzy logika si získala popularitu koncom osemdesiatych a na začiatku deväťdesiatych rokov, vtedy v Japonsku uviedli prvé aplikácie pomocou fuzzy logiky. Najúspešnejšie aplikácie sú v riadení a v regulácii, existujú však aplikácie v rozpoznávaní obrazov, klasifikácií, rozhodovaní a v ďalších oblastiach [11].

Všetky údaje, ktoré počítače prijímajú, spracúvajú, uchovávajú aj produkujú, sú v konečnom dôsledku iba čísla. Človek sa však málokedy vyjadruje v presných pojmoch – jeho vyjadrovanie je často nepresné. Preto by bolo výhodné, aby počítač vedel spracovávať neurčité údaje, aby vedel použiť kvalitatívne pojmy namiesto kvantitatívneho popisu [7].

Fuzzy logika, založená na teórii fuzzy množín a pravdepodobnosti, bola vyvinutá na aproximatívnu reprezentáciu a usudzovanie s nepresnými informáciami často sa vyskytujúcimi v reálnom svete, čo sa odráža v prirodzenom jazyku. Základnou prednosťou fuzzy logiky je schopnosť matematicky podchytiť informácie vyjadrené slovne. Čiže umožňuje pracovať s nejednoznačnými pojmami, často používanými v ľudskej reči. Fuzzy logika umožňuje počítaču „pochopiť“, čo znamená „veľa, málo, vysoký ...“, na základe porovnania. Prednosťou tejto technológie je aj vyhľadávanie informácií na základe nepresných alebo neúplných údajov aj s možnosťou nájsť chybné uložené informácie. Fuzzy logika bola úspešne aplikovaná vo viacerých oblastiach, ako expertné systémy, získavanie a spájanie znalostí, robenie rozhodnutí, získavanie informácií, a ďalšie [12]

V rámci fuzzy logiky sa zaoberáme základnou teóriou fuzzy množín, logických operácií s fuzzy premennými ako aj priblížením lingvistických premenných.

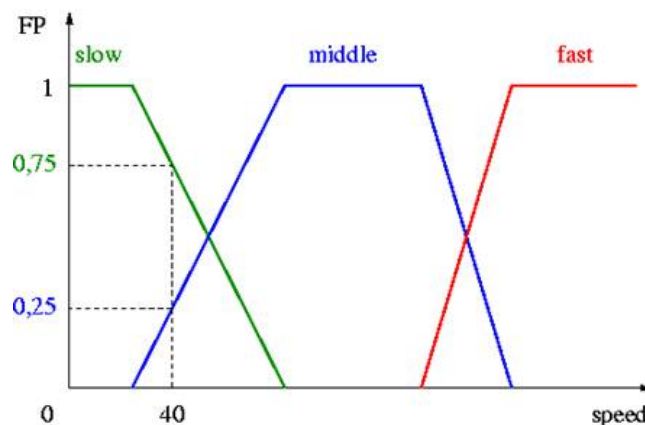
## 2.2 Fuzzy množiny

Pod pojmom fuzzy množiny sa zvyčajne rozumie matematický aparát, ktorý definuje samotný pojem fuzzy množiny a operácie, ktoré možno s fuzzy množinami robiť. Fuzzy množiny sú zovšeobecnením klasických množín.

Ak uvažujeme o klasických množinách, môžeme o každom prvku  $x$  rozhodnúť, že do množiny  $A$  buď patrí (1) alebo nepatrí (0). Príslušnosť prvku  $x$  do fuzzy množiny  $A$  udáva hodnotu funkcie príslušnosti, ktorá môže nadobúdať hodnoty z intervalu  $(0;1)$ . Je potrebné rozlíšiť, že nejde o pravdepodobnosť, s ktorou prvok patrí do fuzzy množiny, ale skôr o silu, s ktorou do nej patrí. Ak charakteristická funkcia nadobúda hodnoty 0 alebo 1, vtedy hovoríme o ostrých množinách. Keď charakteristická funkcia vyjadruje stupeň, s akým prvok do množiny patrí, tak tieto množiny nazývame neostré – fuzzy množiny [7].

Základným problémom pri špecifikácii fuzzy množiny je určenie jej funkcie príslušnosti. Pri formulácii funkcie príslušnosti uvažovanej fuzzy množiny využívame slovné popisy (lingvistické termy), uvádzané zvyčajne expertom v danej oblasti záujmu.

Kvôli lepšiemu porozumeniu jednotlivých výrazov, ktoré boli uvedené, uvádzam príklad (obr 2.1). Ako podklad mi slúžil článok od Jozefa Babjaka s názvom Ako pracujú fuzzy systémy [1]. Majme jeden pojem (tzv. lingvistickú premennú) rýchlosť (speed), ktorú chceme popísať tromi kvalitatívnymi pojmami (tzv. lingvistickými hodnotami): slow, middle a fast. Chceme teda prvok z množiny rýchlosti zaradiť do troch množín, pričom hranice medzi týmito množinami nie sú ostré - preto sú to fuzzy množiny. Každá z týchto množín je definovaná svojou funkciou príslušnosti (FP), tak ako je znázornené na obrázku. Pre rýchlosť 40km/hod môžeme vidieť, že jej príslušnosť do fuzzy množiny "slow" je 0,75, príslušnosť do fuzzy množiny "middle" je 0,25 a príslušnosť do fuzzy množiny "fast" je 0 [1].



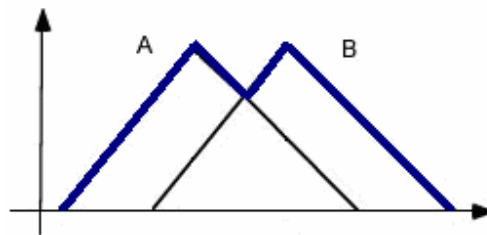
Obr.2.1 Funkcie príslušnosti

Premenná rýchlosť má definované tri hodnoty, ktoré predstavujú tri rôzne fuzzy množiny definované tromi rôznymi lichobežníkovými funkciami príslušnosti, ktorých hodnoty sú na zvislej osi. Funkcie príslušnosti pre jednotlivé fuzzy premenné sú navzájom farebne odlišené. Súčet hodnôt funkcií príslušností v každom bode je rovný jednej; pokiaľ sú funkcie príslušnosti definované týmto spôsobom, hovoríme, že tvoria tzv. fuzzy partíciu. Maximálny počet funkcií príslušností, ktoré sa navzájom prekrývajú, udávajú rád fuzzy partície; funkcie príslušnosti na obrázku tvoria fuzzy partíciu druhého rádu [1], [7].

### 2.3 Operácie s fuzzy množinami

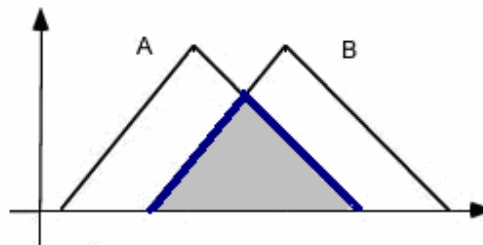
S fuzzy množinami môžeme, podobne ako s klasickými množinami, definovať základné operácie zjednotenie, prienik a doplnok. Význam logických operátorov AND, OR, NOT vo fuzzy logike je rovnaký ako v klasickej booleovskej logike, s tým rozdielom, že v klasickej logike sa viažu na kategorické tvrdenia, ktoré sú buď pravdivé alebo nepravdivé. Vo fuzzy logike sú to limitné stavy pravdivosti [10].

Zjednotenie (logický súčet) dvoch fuzzy množín A a B je fuzzy množina C, ktorá má funkciu príslušnosti [12]:  $m_C = m_{A \vee B}(x) = \max\{m_A(x), m_B(x)\}$



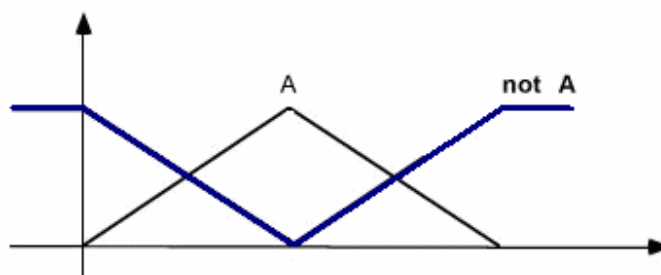
Obr.2.2. Zjednotenie dvoch fuzzy množín (or)

Prienik (logický súčin) dvoch fuzzy množín A a B je fuzzy množina C, ktorá má funkciu príslušnosti [12]:  $m_C = m_{A \wedge B}(x) = \min\{m_A(x), m_B(x)\}$



Obr.2.3. Prienik dvoch fuzzy množín (and)

Doplnok fuzzy množiny A je definovaný pomocou vzťahu [12]:  $m_C(x) = 1 - m_A(x)$

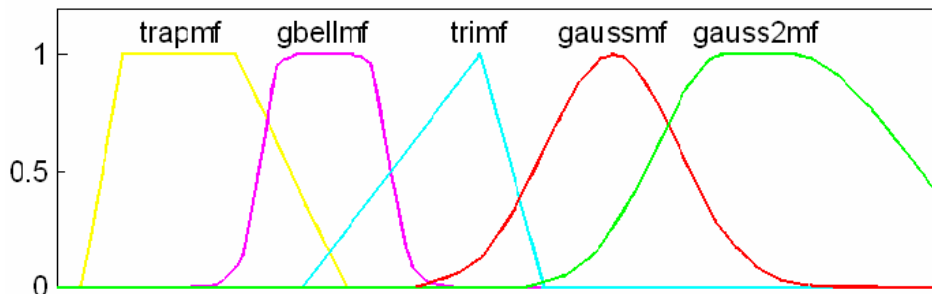


Obr.2.4. Fuzzy množina A a jej doplnok (not)

## 2.4 Funkcia príslušnosti

Funkcia príslušnosti je funkcia, ktorá premení hodnoty vstupného signálu na hodnoty logických premenných [16].

Funkcia príslušnosti charakterizuje stupeň, s ktorým daný prvok do príslušnej množiny patrí. Návrh tvaru, druhu a počtu funkcie príslušnosti býva obyčajne najzložitejšia časť návrhu každého fuzzy systému. Čo sa týka počtu lingvistických hodnôt pre jednu fuzzy premennú, používa sa zvyčajne 3 až 5; viac ako 7 hodnôt zvyčajne človek nedokáže rozlíšiť. Tvar funkcie príslušnosti sa pokiaľ možno používa čo najjednoduchší, kvôli zníženiu výpočtovej náročnosti pri ich vyhodnocovaní; zvyčajne sú to tzv. trojuholníkové alebo lichobežníkové funkcie príslušnosti. Okrem týchto typov poznáme ešte Gaussovu funkciu príslušnosti, zvonovú a sigmoidálnu funkciu príslušnosti. Lichobežníková funkcia príslušnosti je špecifikovaná 4 parametrami. Trojuholníková funkcia príslušnosti je špecifikovaná 3 parametrami, ktoré určujú x-ové súradnice jej troch vrcholov [7], [13].



Obr.2.5. Rôzne tvary funkcie príslušnosti

Špeciálne funkcie príslušnosti nazývané singletony sa používajú najmä na výstupe niektorých fuzzy regulátorov - vyznačujú sa tým, že majú nenulovú hodnotu iba v jednom bode. Pokiaľ uvedené jednoduché druhy funkcie príslušnosti nepostačujú na požadovanú činnosť fuzzy systému, používajú sa "hladké" (spojité a spojitou diferencovateľné) funkcie príslušnosti. Na návrh samotného tvaru funkcie príslušnosti sa používajú rôzne metódy počínajúc odhadom, cez transformáciu frekvenčných a štatistických údajov spracovávaných dát, fyzikálne merania až po metódy založené na adaptácii, učení a ladení parametrov [1].

## 2.5 Lingvistická premenná

Lingvistická premenná je premenná, ktorej hodnoty sú výrazy nejakého jazyka. Hodnotu lingvistickej premennej môžeme interpretovať ako fuzzy – neostre množiny. Lingvistická premenná je usporiadaná päťica  $(L, T(T), U, G, M)$ , kde význam jednotlivých symbolov je nasledujúci [16]:

- $L$  je názov príslušnej premennej,
- $T(T)$  je term - množina, množina všetkých názvov lingvistických hodnôt premennej  $T$ , napr. malá kladná, nulová, veľká kladná ...,
- $U$  je univerzum, na ktorom sú definované jednotlivé termy,
- $G$  je syntetické pravidlo, podľa ktorého je zostavený názov lingvistickej premennej
- $M$  je sémantické pravidlo, miera súhlasu pojmov s jeho významom, ktoré priraduje každej premennej  $x$  jej význam  $M(x)$ .

Pre označenie hodnoty lingvistickej premennej sa používa štandardné označenie. Typické označenie termu – fuzzy hodnôt a ich skratky sú uvedené v tabuľke 1 [10].

Význam	Slovenské označenie	Anglické označenie
Hodnota veľká záporná	ZV	NL
Hodnota stredná záporná	ZS	NM
Hodnota malá záporná	ZM	NS
Hodnota záporná blízka nule	ZN	NZ
Hodnota nulová	N	Z
Hodnota kladná blízka nule	KN	PZ
Hodnota malá kladná	KM	PS
Hodnota stredná kladná	KS	PM
Hodnota veľká kladná	KV	PL

Tab.1 Skratky fuzzy hodnôt

## 2.6 Inferenčné pravidlá

Samotná činnosť fuzzy systému je založená na odvodzovacích (inferenčných) pravidlách. Fuzzy pravidlo možno vyjadriť :

If **x** je **A** then **y** je **B**

Kde **x** a **y** sú lingvistické premenné a **A** a **B** sú lingvistické hodnoty určené fuzzy množinami. Z toho vyplýva, že fuzzy pravidlá sa vzťahujú na fuzzy množiny. Vo fuzzy systémoch je predpoklad pravidla pravdivý v určitej miere a následok je tiež pravdivý v rovnakej miere.

Prvý fuzzy výrok sa nazýva antecedent a druhý fuzzy výrok je konsekvent. Antecedent je podmienková časť pravidla (fuzzy predpoklad). Miera splnenia fuzzy podmienok ovplyvňuje mieru aplikovateľnosti dôsledkovej časti, konsekventu. Vstupy sú obsiahnuté v podmienkovej časti pravidiel, výstupy sú obsiahnuté v konsekvente [10].

## 2.7 Fuzzy model

Väčšina procesov chemickej technológie sú procesy zložité, nelineárne. Napriek tomu, že je známe množstvo rôznych nelineárnych identifikačných metód, v posledných desaťročiach sa v tejto oblasti stále viac uplatňujú prístupy, ktoré vedú k tvorbe fuzzy modelov. Matematický model, vytvorený pomocou fuzzy množín, sa volá fuzzy model. V závislosti od štruktúry pravidla sa používajú dva hlavné typy modelov: model typu Mamdani a model typu Takagi-Sugeno [8].

### Fuzzy model Mamdaniho typu

Tento model bol prvýkrát použitý v aplikácii riadenia parného motora v kombinácii s kotlom pomocou pravidiel vyprodukovaných skúseným ľudským operátorom. Typické pravidlo používané v tomto modeli má tvar :

If **x** je **A** and **y** je **B** then **z** je **C**

Vstupy a výstupy sú opísané pomocou fuzzy množín [13].

### Fuzzy model Sugenovho typu

Pre mnohé účely, napríklad ako modely dynamických systémov, sú veľmi užitočné fuzzy systémy, ktoré majú trochu odlišný tvar pravidiel od pravidiel Mamdaniho typu. Tieto systémy zaviedol M. Sugeno so spolupracovníkmi, preto sa označujú ako Sugenovské modely.

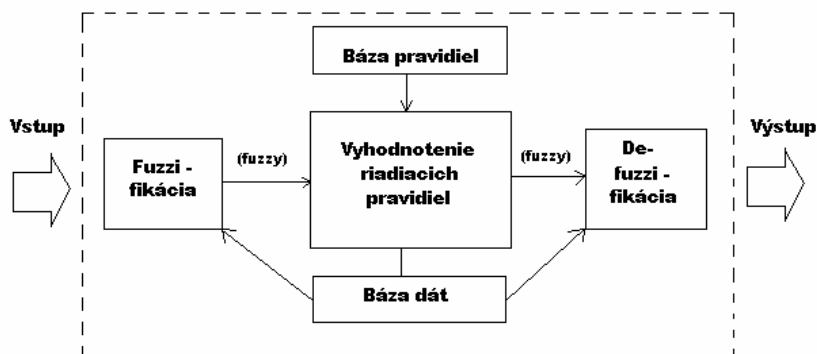
Model typu Sugeno bol navrhnutý Takagim, Sugenom a Kangom ako výsledok snahy o vytvorenie systematického prístupu na generovanie fuzzy pravidiel zo vstupno - výstupných údajov. Typické pravidlo tohto modelu má tvar :

$$\text{If } x \text{ je } A \text{ and } y \text{ je } B \text{ then } z = f(x,y)$$

Ako funkcie  $f(x,y)$  sú často používané jednoduché lineárne kombinácie vstupných premenných. Celkovú výstupnú hodnotu dostaneme ako priemer výstupov pre jednotlivé pravidlá. Sugenovské modely možno veľmi jednoducho použiť najmä pre aproximáciu nelineárnych závislostí. Práca s nimi je pohodlná, najmä pokiaľ poznáme príslušné pravidlá, čo je predpoklad, že sme už spravili rozklad stavového priestoru na príslušný počet buniek charakterizovaných zvolenými fuzzy množinami [16].

## 2.8 Fuzzy inferenčné systémy (FIS)

Základná štruktúra fuzzy inferenčného systému je znázornená na obr.2.6 [14]. Systém pozostáva z piatich základných častí.



Obr.2.6 Fuzzy inferenčný systém (FIS)

Úlohy jednotlivých častí [8], [14], [16]:

➤ **Blok fuzzifikácie**

V bloku fuzzifikácie sú realizované dve operácie, normalizácia a fuzzifikácia. Normalizácia predstavuje prepočet vstupných veličín do normalizovaného univerza. Fuzzifikácia je procedúra meniaci ostré hodnoty normalizovaných vstupných veličín na fuzzy množiny.

➤ **Báza pravidiel**

Báza pravidiel obsahuje všetky pravidlá potrebné pre dostatočnú lingvistickú aproximáciu závislosti medzi vstupom a výstupom regulátora. Jednotlivé pravidlá sú v tvare if-then pravidiel, ktoré majú dve časti :

- premisnú časť výroku (if ...),
- konzekventnú časť výroku (then ...).

Naplniť bázu pravidiel pri návrhu FIS znamená definovať nasledujúce parametre:

- veličiny reprezentujúce stav systému a akčný zásah do systému (jazykové premenné),
- hodnoty jazykových premenných,
- obsah premisných a konzekventných pravidiel,
- a definícia všetkých pravidiel if – then.

➤ **Báza dát**

Dátová báza poskytuje informácie pre správne fungovanie fuzzifikačného, defuzzifikačného bloku a báze pravidiel. Báza dát obsahuje :

- funkcie príslušnosti všetkých vstupných a výstupných fuzzy množín,
- intervaly v ktorých sa pohybujú hodnoty vstupných a výstupných veličín.

Naplnenie tejto bázy znamená, že zadefinujeme funkcie príslušnosti.

➤ **Vyhodnotenie riadiacich pravidiel (inferenčný mechanizmus)**

V tomto bloku sa realizuje inferenčný mechanizmus z rozhodovacích pravidiel, pomocou ktorého získavame zo vstupných fuzzy množín výstupné množiny.

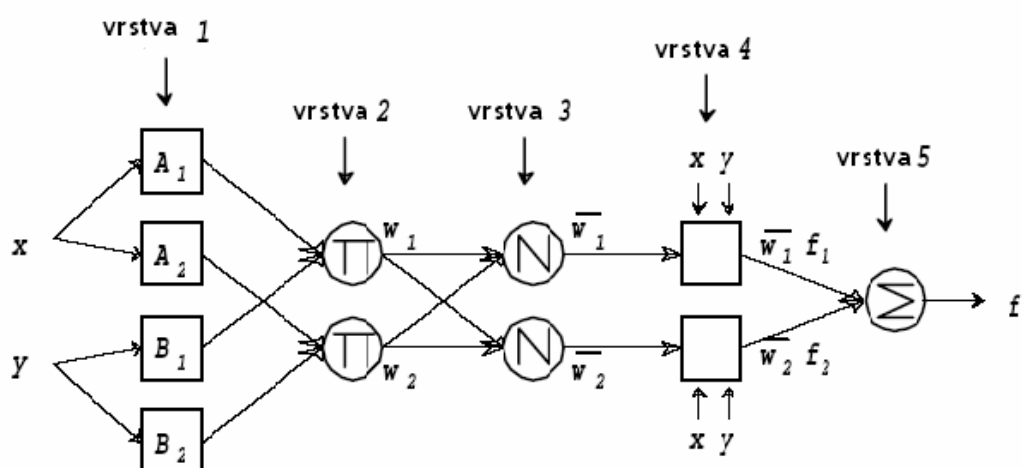
➤ **Blok defuzzifikácie**

Výstup z inferenčného mechanizmu je fuzzy množina, ale akčný zásah do riadeného systému musí byť ostrá hodnota. Preto prvou funkciou defuzzifikačného bloku je vytvoriť z tejto fuzzy množiny ostrú hodnotu, tento proces sa nazýva defuzzifikácia. Táto ostrá hodnota je ale vyjadrená internou reprezentáciou fuzzy regulátora a je treba ho prepočítať na fyzikálny rozmer akčnej veličiny. Tento proces nazývame denormalizáciou.

## 2.9 Tvorba fuzzy modelov metódou Anfis

Jednou z možností ako odvodiť bázu podmienených pravidiel z historických údajov, je tzv. Adaptive Neuro Fuzzy Inference System metóda (ANFIS metóda). Podstatou tejto metódy je neuro – adaptívny proces učenia, na základe ktorého možno z historických údajov odvodiť parametre funkcií príslušnosti a extrahovať bázu pravidiel. Pomocou tejto metódy sa priradujú vstupné údaje k výstupným údajom, pričom v procese učenia sa postupne menia parametre jednotlivých funkcií príslušnosti tak, aby sa čo najlepšie charakterizovali vzťahy medzi priestorom vstupných premenných a priestorom výstupných premenných. V procese učenia možno použiť metódu spätného šírenia chyby, kombináciu metódy spätného šírenia chyby s metódou najmenších štvorcov, resp. evolučné stochastické optimalizačné algoritmy [15].

Neuro – fuzzy systém ANFIS (obr.2.7) svojou činnosťou zodpovedá Takagi – Sugeno fuzzy regulátoru. Vnútorne je ale reprezentovaný ako päťvrstvová dopredná neurónová sieť, ktorej učenie je založené na minimalizácii odchýlok metódou najmenších štvorcov a zodpovedá úprave tzv. premisných a konsekventných parametrov. Tieto parametre nie sú uložené vo váhach spomínanej neurónovej siete, ale v jej uzloch. Výsledkom ANFISu je fuzzy inferenčný systém typu Takagi – Sugeno.



Obr. 2.7 Architektúra siete Anfis

Vrstvu 1 tvoria adaptívne uzly, každý uzol predstavuje lingvistickú hodnotu vstupnej premennej.

Vrstvu 2 tvoria neadaptívne uzly, ktoré iba násobia prichádzajúce signály a na výstupe tak poskytujú váhu  $w$  pravidiel, ktorých antecedent je daný kombináciou lingvistických hodnôt jednotlivých premenných.

Vrstvu 3 tvoria neadaptívne uzly, ktoré poskytujú na výstupe pomer váhy jednotlivých pravidiel a súčtu váh všetkých pravidiel  $\bar{w}$ .

Vrstvu 4 tvoria adaptívne uzly, ktorých prenosové funkcie sú dané požadovaným tvarom konsekventu. Môžu byť konštantné alebo lineárne.

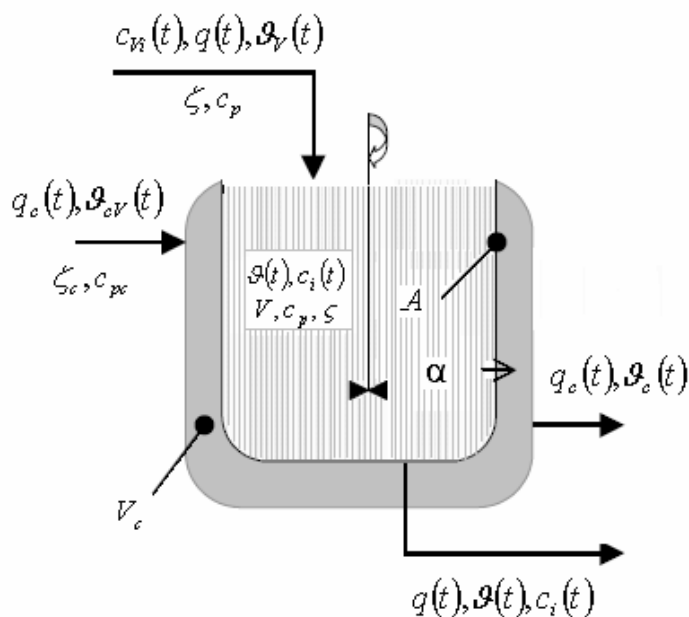
Vrstvu 5 tvorí jeden neadaptívny uzol, ktorý počíta celkový výstup ako súčet všetkých vstupných signálov [2], [15].

### 3 Prietokový chemický reaktor

Chemické reaktory sú súčasťou mnohých technológií v chemickom a spotrebnom priemysle. Je známe, že riadenie procesu v chemických reaktoroch je pomerne náročné vzhľadom k ich vlastnostiam. Dôsledkom toho sú nelinearity a pomerne silné exotermické reakcie.

#### 3.1 Popis chemického reaktora

Uvažujeme o prietokovom chemickom reaktore s chladením v plášti. Schéma reaktora je znázornená na obr. 3.1. Teplota reakčnej zmesi sa riadi prietokom chladiaceho média. Žiadaná teplota reakčnej zmesi je  $\vartheta = 359$  K. V reaktore ako chladiace médium použijeme vodu. Teplota chladiaceho média na vstupe je  $\vartheta_{cV} = 298$  K a prietok  $q_c = 0,004 \text{ m}^3 \text{ min}^{-1}$ . Vstupná teplota reakčnej zmesi je  $\vartheta_V = 328$  K a prietok reakčnej zmesi je  $q = 0,015 \text{ m}^3 \cdot \text{min}^{-1}$ . Koncentrácie jednotlivých zložiek na vstupe je nasledovné:  $c_{VA} = 4,22 \text{ kmol} \cdot \text{m}^{-3}$ ,  $c_{VB} = 0 \text{ kmol} \cdot \text{m}^{-3}$ ,  $c_{VC} = 0 \text{ kmol} \cdot \text{m}^{-3}$ . Hustota reakčnej zmesi  $\rho = 1020 \text{ kg} \cdot \text{m}^{-3}$  a chladiaceho média je  $\rho_c = 998 \text{ kg} \cdot \text{m}^{-3}$ . Špecifická tepelná kapacita reakčnej zmesi je  $C_p = 4,02 \text{ kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$  a chladiaceho média je  $C_{pc} = 4,182 \text{ kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$ . Pre riadenie je potrebné poznať aj parametre reaktora. Objem reakčnej zmesi je  $V = 0,23 \text{ m}^3$  a objem chladiaceho média je  $V_c = 0,21 \text{ m}^3$ . Plocha prestupu tepla z reakčnej zmesi do chladiaceho média je  $A = 1,51 \text{ m}^2$  a úhrnný koeficient prechodu tepla je  $\alpha = 42,8 \text{ kJ} \cdot \text{min}^{-1} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$ . Dôležitou úlohou je opísať aj odohrávajúce sa chemické reakcie. Podiel aktivačnej energie a univerzálnej plynovej konštanty 1. reakcie je  $g_1 = 9850$  K. Podiel aktivačnej energie a univerzálnej plynovej konštanty 2. reakcie je  $g_2 = 22019$  K. Reakčné entalpie sú  $(\Delta_r H)_1 = -8,6 \cdot 10^4 \text{ kJ} \cdot \text{kmol}^{-1}$ ,  $(\Delta_r H)_2 = -1,82 \cdot 10^4 \text{ kJ} \cdot \text{kmol}^{-1}$ . Pri výpočte rýchlostných konštánt je potrebné poznať hodnoty predexponenciálnych faktorov. Hodnoty sú nasledovné  $k_{1,\infty} = 1,55 \cdot 10^{11} \text{ min}^{-1}$  a  $k_{2,\infty} = 4,55 \cdot 10^{25} \text{ min}^{-1}$ .



Obr.3.1. Prietokový chemický reaktor

Predpokladáme, že reakčná zmes v reaktore aj chladiaca kvapalina v plášti sú dokonale miešané. Tepelnú kapacitu steny oddeľujúcej reakčný priestor od chladiacej kvapaliny zanedbáme. Objem reakčnej zmesi v reaktore, objem chladiaceho média, hustotu a tepelné kapacity reakčnej zmesi a chladiča ako aj koeficient prestupu tepla v reaktore pokladáme za konštantnú. Tlak v systéme je konštantný, prestup tepla uskutočňuje len prúdením. Straty tepla do okolia sú zanedbateľné. Predpokladáme, že všetky reakcie sú exotermické, teda ich reakčné entalpie sú záporné, t.j.  $(\Delta_r H)_j < 0$ . V reaktore sa odohrávajú dve paralelné reakcie :



Rýchlostné konštanty  $k_1$  a  $k_2$  vypočítame nasledovne :

$$k_1(t) = k_{1,\infty} e^{\frac{E_1}{Rv}} \quad k_2(t) = k_{2,\infty} e^{\frac{E_2}{Rv}} \quad (3.1.3)$$

Pomocou týchto predpokladov môžeme odvodiť dynamický matematický model reaktora.

### 3.2 Dynamický matematický model reaktora

Dynamický matematický model tvoria materiálové bilancie reagujúcich zložiek, entalpická bilancia reakčnej zmesi a entalpická bilancia chladiaceho média [3].

#### Materiálová bilancia zložky A

$$q(t)c_{vA}(t) + \nu_{A1}\dot{\xi}_{V1}(t)V + \nu_{A2}\dot{\xi}_{V2}(t)V = q(t)c_A(t) + \frac{d[Vc_A(t)]}{dt} \quad (3.2.1)$$

Veličina  $\dot{\xi}_{Vj}(t)$  je okamžitá objemová rýchlosť  $j$ -tej chemickej reakcie. V rovnici (3.2.2) sú znázornené výpočtové rovnice pre túto veličinu.

$$\dot{\xi}_{V1} = \frac{d\xi_{V1}(t)}{dt} = \frac{dn_{A1}(t)}{\nu_{A1}dt} \frac{1}{V} \quad \dot{\xi}_{V2} = \frac{d\xi_{V2}(t)}{dt} = \frac{dn_{A2}(t)}{\nu_{A1}dt} \frac{1}{V} \quad (3.2.2)$$

Rýchlostnú rovnicu môžeme vyjadriť aj vo všeobecnej forme.

$$\dot{\xi}_{Vj}(t) = k_j(t)f(c_i, i=1, \dots, n) \quad (3.2.3)$$

Rýchlostné konštanty  $k_j(t)$  už boli odvodené, vid' rovnice (3.1.3).

Po úprave a dosadení všetkých konštánt rovnica (3.2.1) nadobudne nasledujúci tvar.

$$\frac{dc_A(t)}{dt} = \frac{q(t)}{V}(c_{vA}(t) - c_A(t)) - k_1(t)c_A(t) - k_2(t)c_A(t) \quad (3.2.4)$$

#### Materiálová bilancia zložky B

$$q(t)c_{vB}(t) - \nu_{B1}\dot{\xi}_{V1}(t)V = \frac{d[Vc_B(t)]}{dt} + q(t)c_B(t) \quad (3.2.5)$$

Koncentrácia zložky B na vstupe do reaktora je nulová, preto rovnicu (3.2.5) upravíme.

Pri úprave vyjadríme aj objemovú rýchlosť .

$$\frac{dc_B(t)}{dt} = k_1(t)c_A(t) - \frac{q(t)c_B(t)}{V} \quad (3.2.6)$$

### Materiálová bilancia zložky C

$$\frac{dc_C(t)}{dt} = k_2(t)c_A(t) - \frac{q(t)c_C(t)}{V} \quad (3.2.7)$$

### Entalpická bilancia reakčnej zmesi

$$\begin{aligned} \frac{d\mathcal{G}(t)}{dt} = & \frac{q(t)[\mathcal{G}_v(t) - \mathcal{G}(t)]}{V} + \frac{k_1c_A(t)(-\Delta_r H)_1}{\rho c_p} + \\ & + \frac{k_2c_A(t)(-\Delta_r H)_2}{\rho c_p} - \frac{A\alpha[\mathcal{G}(t) - \mathcal{G}_c(t)]}{V\rho c_p} \end{aligned} \quad (3.2.8)$$

### Entalpická bilancia chladiaceho média

$$\frac{d\mathcal{G}_c(t)}{dt} = \frac{q_c(t)\mathcal{G}_{cv}(t)}{V_c} - \frac{q_c(t)\mathcal{G}_c(t)}{V_c} + \frac{A\alpha[\mathcal{G}(t) - \mathcal{G}_c(t)]}{V_c\rho_c c_{p_c}} \quad (3.2.9)$$

### Počiatočné podmienky

$$\begin{aligned} c_A(t_0) &= c_A^s & \mathcal{G}(t_0) &= \mathcal{G}^s \\ c_B(t_0) &= c_B^s & \mathcal{G}_c(t_0) &= \mathcal{G}_c^s \\ c_C(t_0) &= c_C^s \end{aligned} \quad (3.2.10)$$

Stavové veličiny sú :  $c_A(t)$ ,  $c_B(t)$ ,  $c_C(t)$ ,  $\mathcal{G}(t)$ ,  $\mathcal{G}_c(t)$

### 3.3 Matematický model rovnovážneho stavu a analýza stability

Predpokladom na sledovanie dynamických vlastností systému je výpočet ustáleného stavu. Proces je v ustálenom stave, ak sa vstupné a výstupné veličiny v čase  $t$  nemenia. V rovnovážnom stave nedochádza k akumulácii [3].

V ustálenom stave tvar materiálových bilancií jednotlivých zložiek a entalpických bilancií je nasledovný :

Materiálová bilancia zložky A

$$q^s c_{VA}^s - k_1^s c_A^s V - k_2^s c_A^s V = q^s c_A^s \quad (3.3.1)$$

Materiálová bilancia zložky B

$$k_1^s c_A^s V = q^s c_B^s \quad (3.3.2)$$

Materiálová bilancia zložky C

$$k_2^s c_A^s V = q^s c_C^s \quad (3.3.3)$$

Entalpická bilancia reakčnej zmesi

$$q^s \rho c_p \mathcal{G}_V^s + k_1^s c_A^s V (-\Delta_r H)_1 + k_2^s c_A^s V (-\Delta_r H)_2 = q^s \rho c_p \mathcal{G}^s + A \alpha [\mathcal{G}^s - \mathcal{G}_c^s] \quad (3.3.4)$$

Entalpická bilancia chladiaceho média

$$q_c^s \rho_c c_{pC} \mathcal{G}_{VC}^s + A \alpha [\mathcal{G}^s - \mathcal{G}_c^s] = q_c^s \rho_c c_{pC} \mathcal{G}_C^s \quad (3.3.5)$$

Ustálený stav som vypočítala pomocou týchto 5 rovníc v Matlabe príkazom *fsolve*.. Hodnoty stavových veličín v ustálenom stave sú uvedené v tab.2.

$c_A^s$ [ kmol.m <sup>-3</sup> ]	$c_B^s$ [ kmol.m <sup>-3</sup> ]	$c_C^s$ [ kmol.m <sup>-3</sup> ]	$\vartheta^s$ [K]	$\vartheta_c^s$ [K]
0,49	2	1,72	363,6	350,1

Tab.2 Hodnoty jednotlivých veličín v ustálenom stave

Po určení rovnovážneho stavu reaktora je potrebné určiť, či tento stav je stabilný. Pre analýzu stability reaktora je rozhodujúca tepelná bilancia reakčnej zmesi (3.3.4). Túto rovnicu upravíme do tvaru :

$$k_1^s c_A^s V(-\Delta_r H)_1 + k_2^s c_A^s V(-\Delta_r H)_2 = q\rho c_p [\vartheta^s - \vartheta_v] + A\alpha [\vartheta^s - \vartheta_c^s] \quad (3.3.6)$$

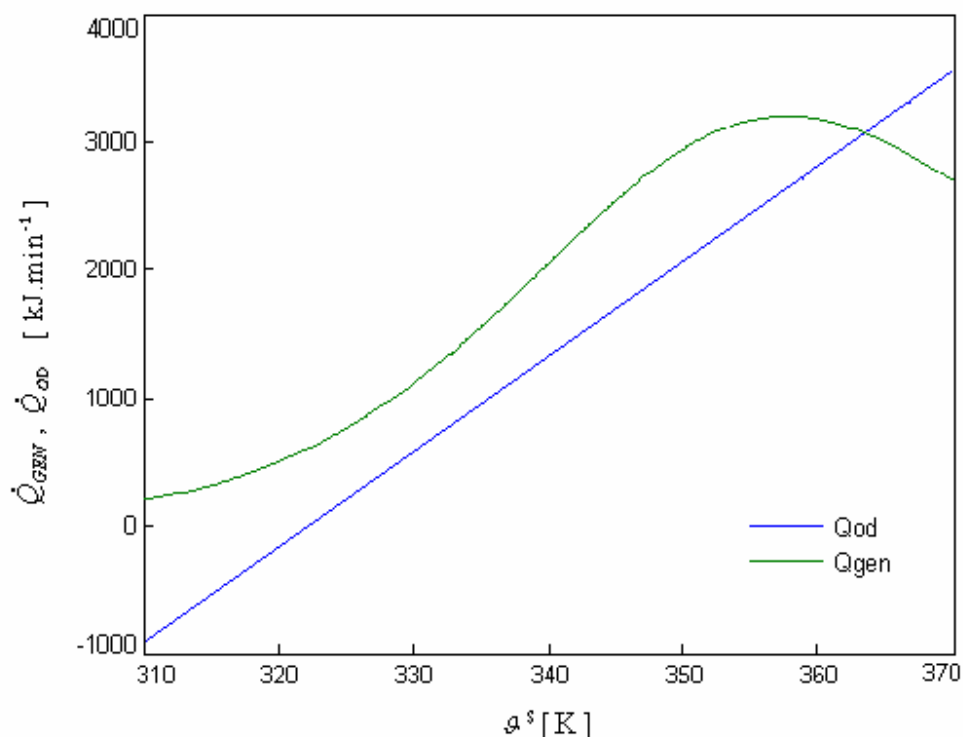
Ľavá strana tejto rovnice vyjadruje rýchlosť tvorby tepla v systéme. Rovnica vygenerovaného tepla (3.3.7) je nelineárnou funkciou teploty reakčnej zmesi v rovnovážnom stave.

$$\dot{Q}_{GEN} = k_1^s c_A^s V(-\Delta_r H)_1 + k_2^s c_A^s V(-\Delta_r H)_2 \quad (3.3.7)$$

Pravá strana rovnice (3.3.6) vyjadruje rýchlosť odvádzania tepla zo systému. Z rovnice (3.3.8) vyplýva, že rýchlosť odovzdávania tepla je lineárnou funkciou teploty reakčnej zmesi v rovnovážnom stave.

$$\dot{Q}_{OD} = q\rho c_p [\vartheta^s - \vartheta_v] + A\alpha [\vartheta^s - \vartheta_c^s] \quad (3.3.8)$$

Chemický reaktor je v rovnovážnom stave vtedy, ak rýchlosť tvorby tepla v systéme sa rovná rýchlosti odovzdávania tepla. Správnosť výpočtu ustáleného stavu som overila pomocou analýzy, výsledok je znázornený na obr.3.2.



Obr. 3.2 Analýza stability reaktora

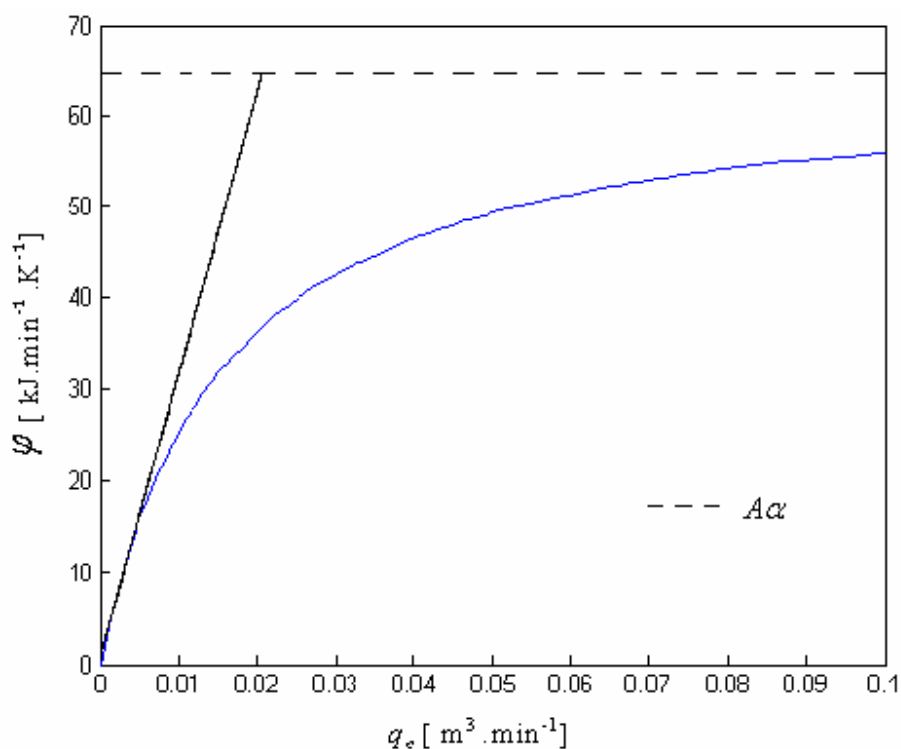
Priesečníkom krivky rýchlosti tvorby tepla chemickými reakciami a priamky rýchlosti odvádzania tepla zo systému je jeden bod, tento bod vyjadruje rovnovážny stav. Z analýzy vyplýva, že reaktor v rovnovážnom stave je stabilný.

### 3.4 Analýza prietoku chladiaceho média

Prietok chladiaceho média je akčnou veličinou, preto je potrebné vedieť, do akej miery je možné zvýšiť jeho hodnotu. Pre analýzu vplyvu prietoku chladiaceho média na chladenie reaktora je rozhodujúca tepelná bilancia chladiaceho média. Úpravy musíme robiť tak, aby sme tepelný tok, ktorý prestupuje z reakčnej zmesi do chladiaceho média, vyjadrili pomocou prietoku chladiaceho média. Z rovnice (3.3.5) najprv vyjadríme teplotu chladiaceho média v ustálenom stave, ktorý potom dosadíme do člena reprezentujúceho tepelný tok, ktorý prestupuje z reakčnej zmesi do chladiaceho média. V rovnici (3.3.9) je uvedená už upravená rovnica tepelného toku.

$$\begin{aligned}\dot{Q}_C &= A\alpha[\vartheta^s - \vartheta_c^s] = A\alpha \left[ \vartheta^s - \frac{q_c^s \rho_c c_{pC} \vartheta_{VC}^s + A\alpha \vartheta^s}{q_c^s \rho_c c_{pC} + A\alpha} \right] = \\ &= \frac{A\alpha q_c^s \rho_c c_{pC}}{q_c^s \rho_c c_{pC} + A\alpha} (\vartheta^s - \vartheta_{VC}^s) = \varphi(\vartheta^s - \vartheta_{VC}^s)\end{aligned}\quad (3.3.9)$$

Funkcia  $\varphi$  je nelineárnou funkciou prietoku  $q_c$ , teda aj tepelný tok. Na získanie maximálnej hodnoty prietoku chladiaceho média potrebujeme nakresliť závislosti tepelného toku od prietoku chladiaceho média [3]. Závislosť je znázornená na obr. 3.3.



Obr. 3.3 Analýza vplyvu chladiaceho média na chladenie reaktora

Z grafického znázornenia vyplýva, že prietok chladiaceho média má význam zvyšovať len po hodnotu  $0,02 \text{ m}^3 \cdot \text{min}^{-1}$ .

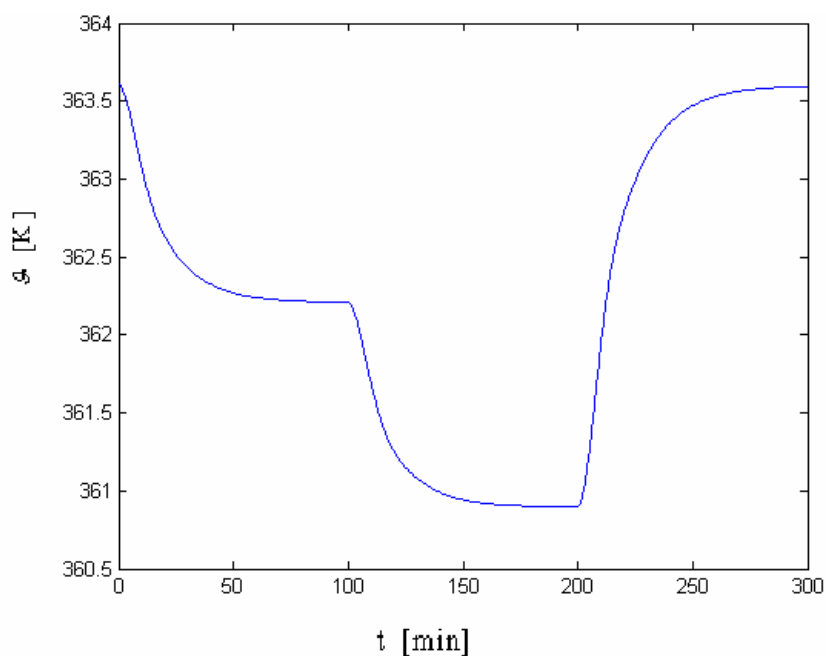
## 4 Vytvorenie fuzzy modelov

Väčšina procesov v chemickej technológii sú procesy zložité, nelineárne. Napriek tomu, že poznáme metódy na linearizáciu týchto procesov, v poslednom čase sa v tejto oblasti stále viac využíva vytváranie matematických modelov pomocou fuzzy množín.

### 4.1 Návrh fuzzy modelu Takagi – Sugenovho typu

Najčastejšie používanými fuzzy modelmi sú modely typu Mamdani a Takagi – Sugeno (T-S).

V tejto kapitole je odvodený návrh fuzzy modelu Takagi – Sugenovho typu. Vstupno-výstupné údaje na návrh takéhoto fuzzy modelu získame simuláciou chemického reaktora s-funkciou. Pri simulácii je vstupnou veličinou prietok chladiaceho média a výstupnou veličinou je teplota reakčnej zmesi. Prietok chladiaceho média bol skokom zmenený z hodnoty  $q_c = 0,004 \text{ m}^3\text{min}^{-1}$  na  $q_c = 0,005$ ;  $0,006$  a  $0,004 \text{ m}^3\text{min}^{-1}$ . Skokové zmeny boli zavedené v čase  $t = 0,100$  a  $200 \text{ min}$ .

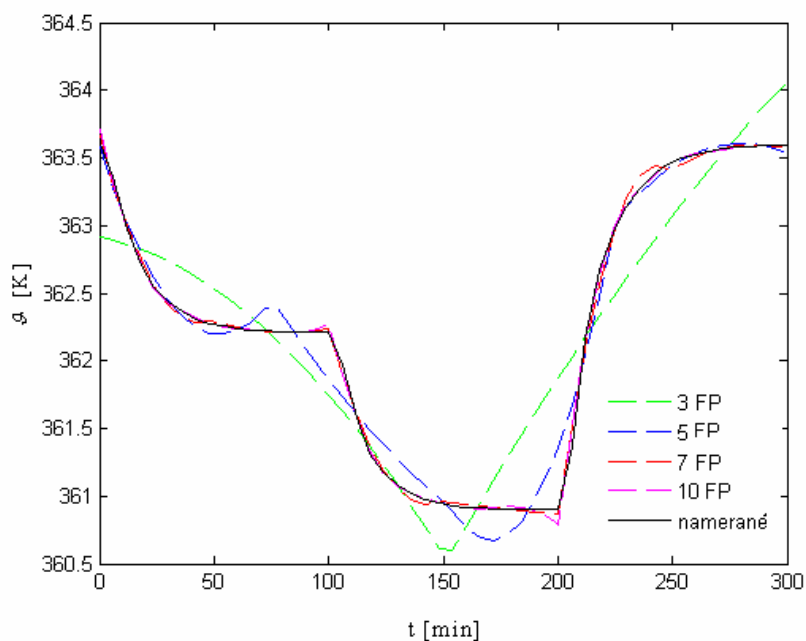


Obr. 4.1 Prechodová charakteristika reaktora

Pri projektovaní fuzzy modelov je vstupnou veličinou čas a výstupnou veličinou sú namerané teploty reakčnej zmesi. Ďalším krokom je inicializácia fuzzy systému, na ktorú použijeme príkazy `genfis1` a `genfis2`.

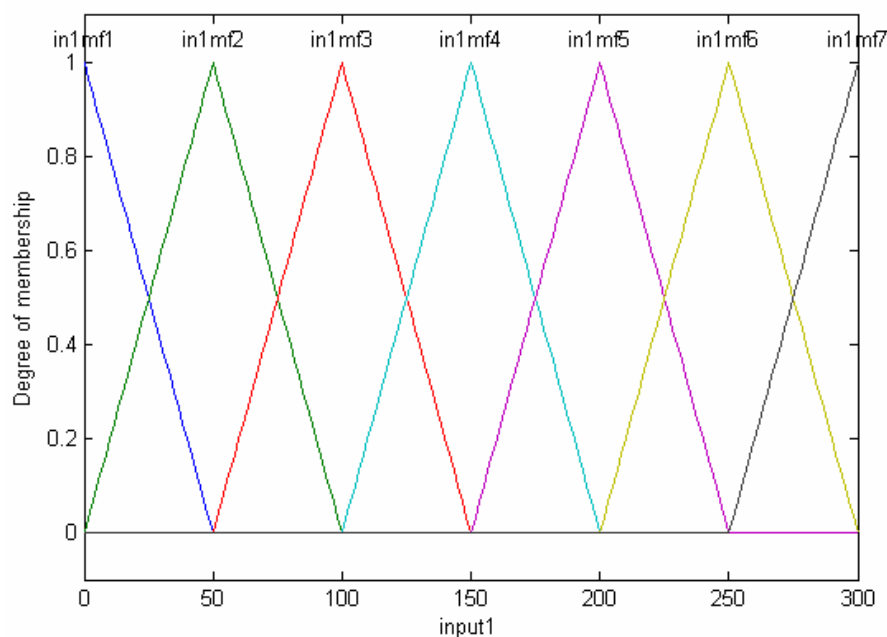
#### 4.1.1 Inicializácia pomocou príkazu `genfis1`

Nato, aby sme mohli použiť príkaz `genfis1`, potrebujeme zadať typ a počet funkcií príslušnosti. Na obr. 4.2 je znázornený vplyv počtu funkcií príslušnosti na priebeh teploty reakčnej zmesi. Na porovnanie boli použité funkcie príslušnosti trojuholníkového tvaru.



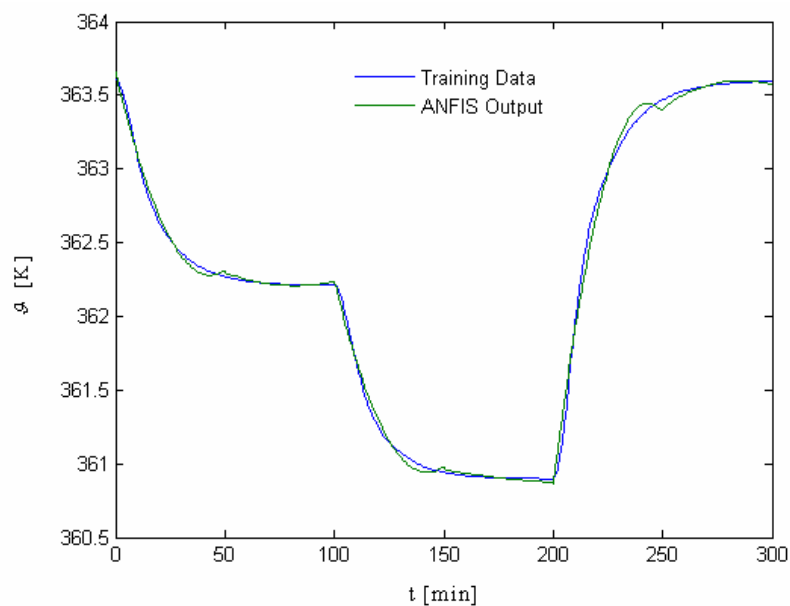
Obr. 4.2 Vplyv počtu funkcií príslušnosti na priebeh teploty reakčnej zmesi

Z porovnania vyplýva, že na návrh presného modelu je výhodnejšie použiť vyšší počet funkcií príslušnosti. Pri počte funkcií príslušnosti 7 a 10 je rozdiel medzi nameranými hodnotami a fuzzy modelom T-S minimálny. Pri návrhu fuzzy modelu použijeme počet funkcií príslušnosti 7.



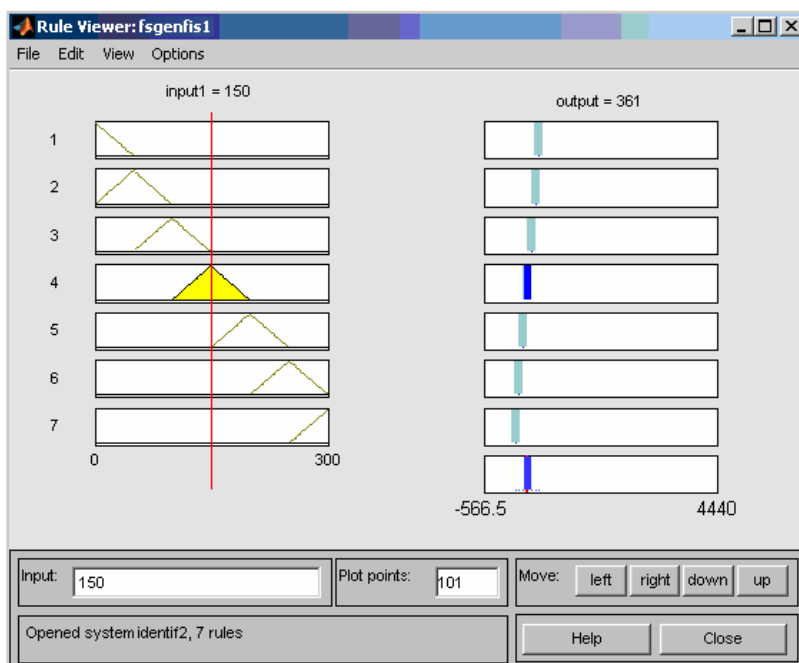
Obr. 4.3 Funkcie príslušnosti fuzzy modelu T-S

Po vybraní počtu a typu funkcií príslušnosti (obr. 4.3 ) nasleduje inicializácia fuzzy systému príkazom `genfis1`. `Genfis1` generuje výstup fuzzy inferenčného systému (FIS) typu Sugeno pomocou sieťového rozdelenia dát. Tento FIS poskytuje počiatočné podmienky pre `Anfis` tréning. Dáta pre inicializáciu sú v tvare matíc s počtom stĺpcov  $n+1$ , kde stĺpec  $n$  obsahuje dáta pre každý FIS vstup a posledný stĺpec je FIS výstup. Každé pravidlo vygenerované s `genfisom1` má jednu funkciu príslušnosti výstupu. Príkaz `genfis1` v Matlabe píšeme v tvare: `in_fismat = genfis1(trnData,numMFs,mfType)`. `TrnData` predstavuje vstup a výstup v tvare matíc, `numMFs` je počet funkcií príslušnosti a `mfType` je typ funkcií príslušnosti. Po inicializácii nasleduje tréning siete pomocou `Anfis`, ktorý používa hybridnú učiacu metódu. Na základe kombinácie metódy spätného šírenia chyby s metódou najmenších štvorcov sa trénujú funkcie príslušnosti parametrov FIS na modelovanie vstupných a výstupných dát. Príkaz `Anfis` v Matlabe píšeme v tvare : `out_fismat = anfis(trnData,in_fismat,3)`. Obr. 4.4 znázorňuje porovnanie dát, ktoré sú určené na tréning a výstupné údaje z `Anfis`.



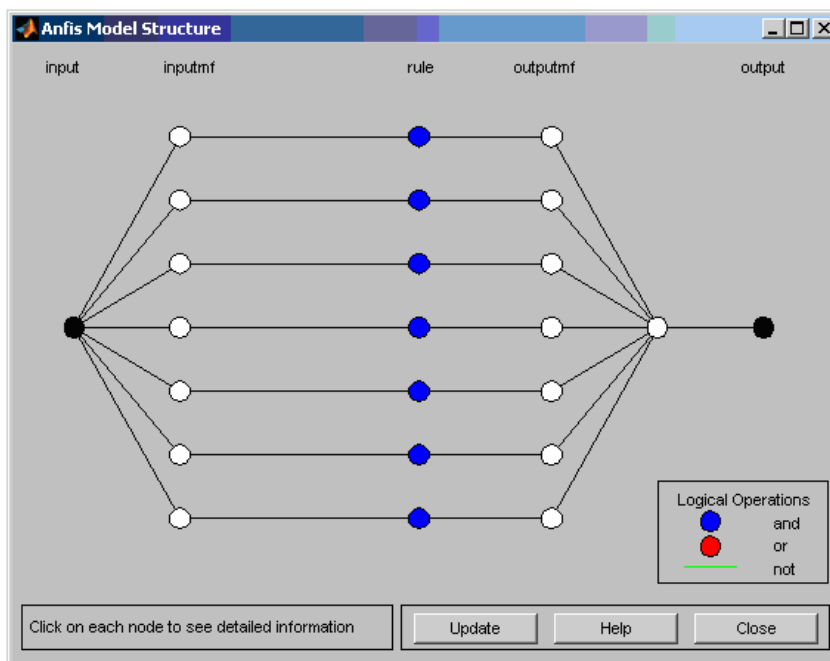
Obr. 4.4 Porovnanie výstupu z Anfis štruktúry

*Rule viewer* je grafické zobrazenie procesu inferencie. V tomto vieweri sú zobrazené všetky pravidlá, tvary funkcií príslušnosti vstupov a výstupov a ich inferencie.



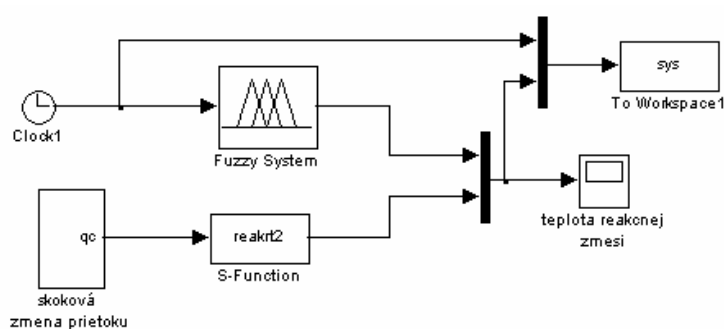
Obr. 4.5 Grafické zobrazenie procesu inferencie

Pomocou červenej čiary si vieme pozrieť výstupy pri určitých ostrých hodnotách. Výslednú štruktúru Anfís (obr. 4.6) môžeme graficky znázorniť pomocou príkazu *anfisedit*.



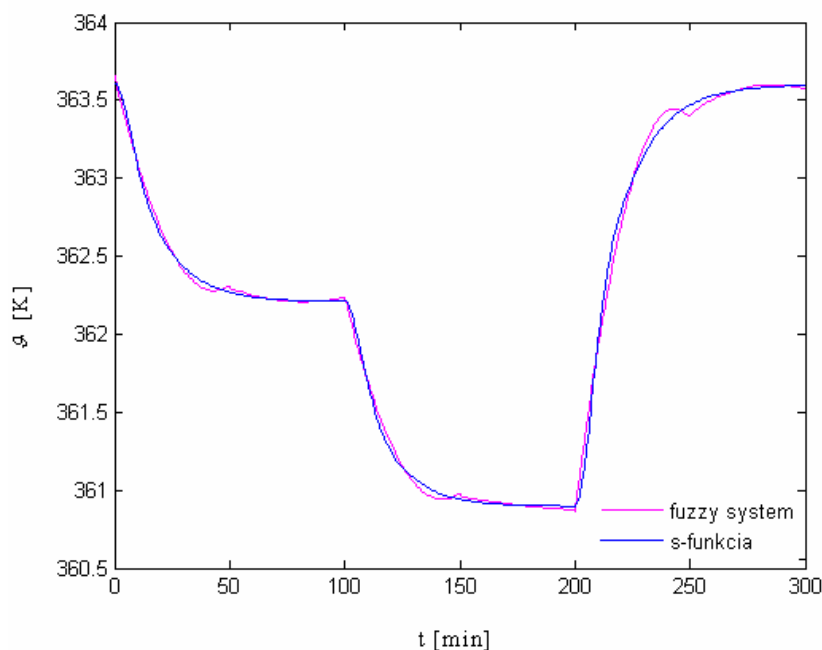
Obr. 4.6 Výsledná štruktúra Anfís

Na overenie správnosti návrhu fuzzy modelu použijeme nasledovnú blokovú schému:



Obr. 4.7 Bloková schéma na simuláciu

Výsledok simulácie je znázornený na obr. 4.8. Z obrázka vyplýva, že rozdiel medzi prechodovou charakteristikou reaktora pomocou s-funkcie a fuzzy modelu (*fsgenfis1.fis*) je minimálny.

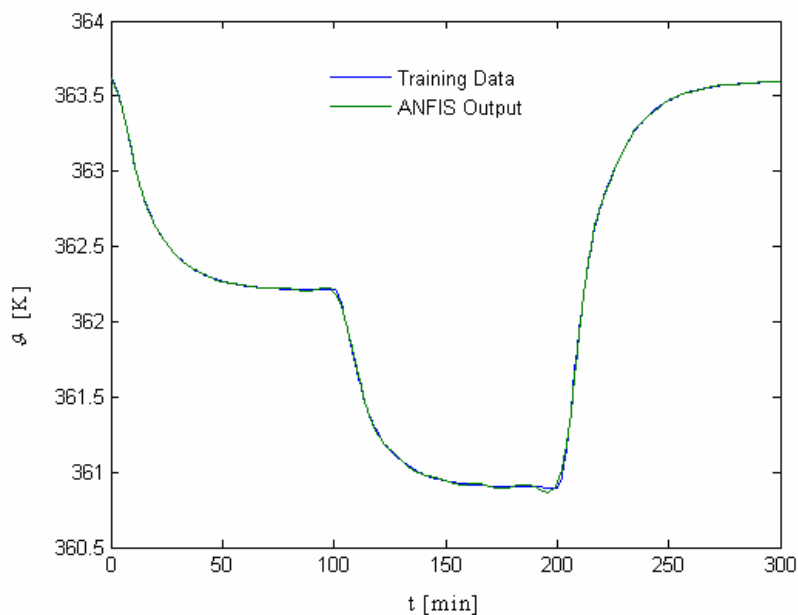


Obr. 4.8 Priebeh teploty reakčnej zmesi

Program použitý na návrh a overenie správnosti fuzzy modelu, je uvedený v prílohe.

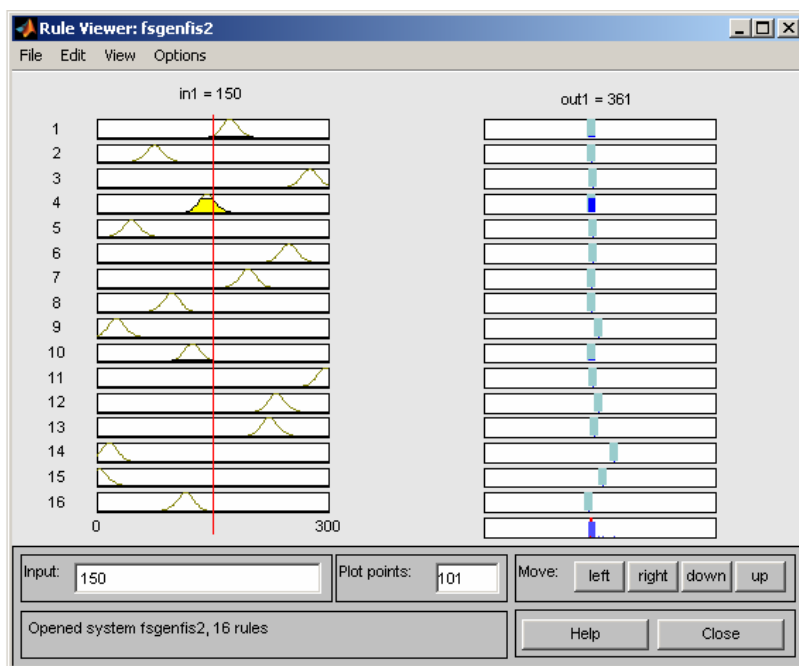
#### 4.1.2 Inicializácia pomocou príkazu *genfis2*

Inicializáciu fuzzy systému môžeme vykonať aj pomocou príkazu *genfis2*. Vstupné a výstupné údaje sú totožné s údajmi použitými v sekcii 4.1.2. Fuzzy inferenčný systém je vytvorený metódou subtraktívneho zhukovania. Príkaz *genfis2* v Matlabe píšeme v tvare: *out\_fismat = genfis2(x,y,0.1)*. Kde *x* a *y* sú vstupy a výstupy. Číslo 0.1 špecifikuje rád vplyvu v centre zhuku. Rozsah tohto rádu je od 0 do 1. Obr. 4.9 znázorňuje dáta, ktoré slúžili na tréovanie a dáta vystupujúce z Anfis štruktúry.

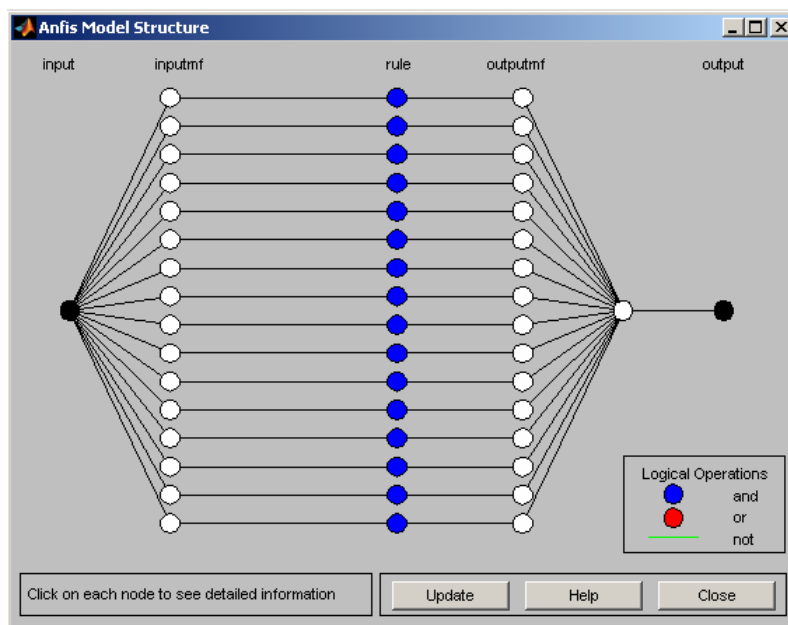


Obr.4.9 Porovnanie výstupu z Anfis štruktúry

V *rule viewer* (obr. 4.10), sú zobrazené všetky pravidlá, tvary funkcií príslušnosti vstupov, výstupov a ich inferencie. Pri návrhu príkazom *genfis2* použijeme 16 funkcií príslušnosti typu *gaussmf*. Obr. 4.11 znázorňuje výslednú Anfis štruktúru.

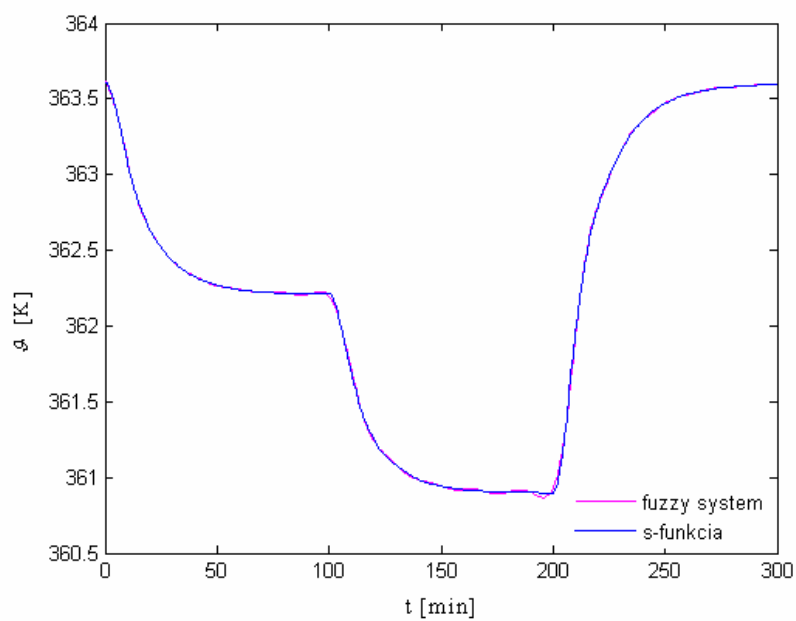


Obr. 4.10 Grafické zobrazenie procesu inferencie



Obr. 4.11 Výsledná Anfis štruktúra

Pomocou grafického porovnania (obr. 4.12) môžeme usúdiť, že prechodové charakteristiky simulované pomocou s-funkcie a fuzzy systémom (*fsgenfis2.fis*) sú totožné.



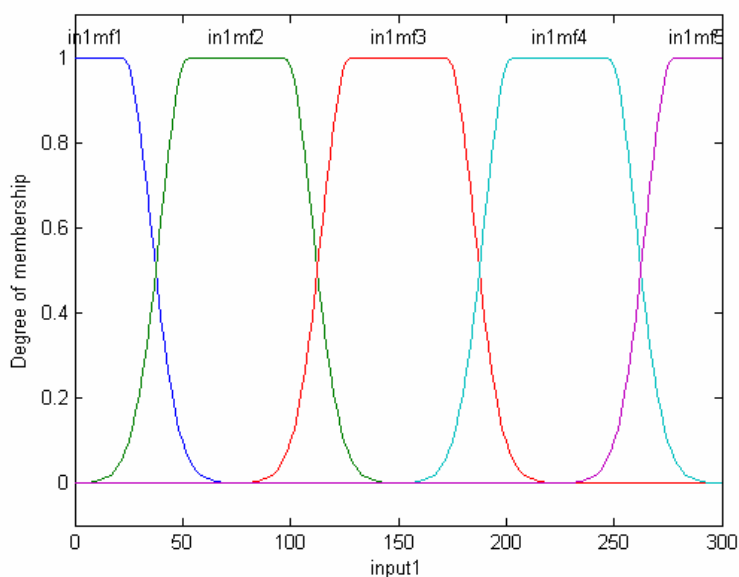
Obr. 4.12 Porovnanie prechodových charakteristik

Program používaný na návrh a overenie správnosti návrhu fuzzy modelu (*fsgenfis2.fis*) je uvedený v prílohe.

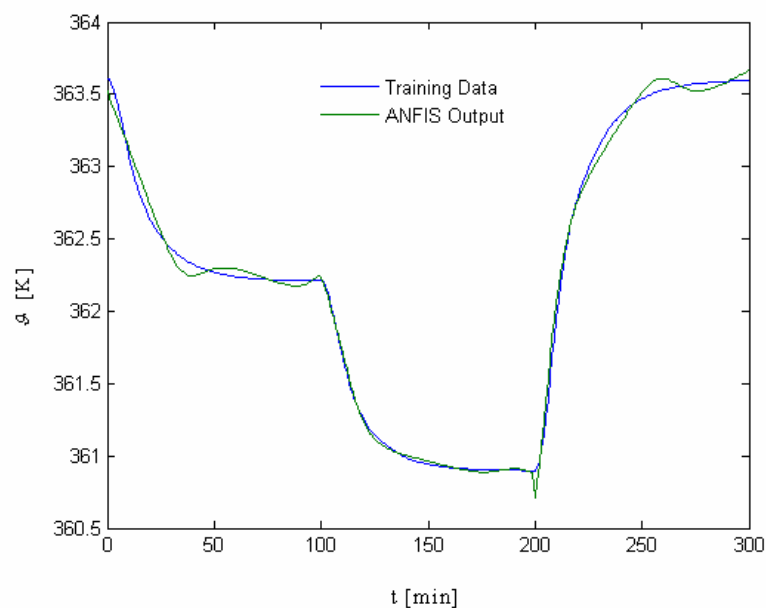
## 4.2 Návrh fuzzy modelu pre viac vstupných premenných

Uvedené fuzzy modely mali jeden vstup a jeden výstup. Počet vstupných premenných môže byť iný, všeobecne  $N$ .

Pri projektovaní fuzzy modelu sú vstupnou veličinou čas a meniaci sa prietok chladiaceho média. Výstupnou veličinou sú namerané teploty reakčnej zmesi. Tieto údaje máme zo simulácie chemického reaktora s-funkciou. Na inicializáciu fuzzy systému použijeme príkaz `genfis1` a 5 funkcií príslušnosti s tvarom *gauss2mf* (obr. 4.13).

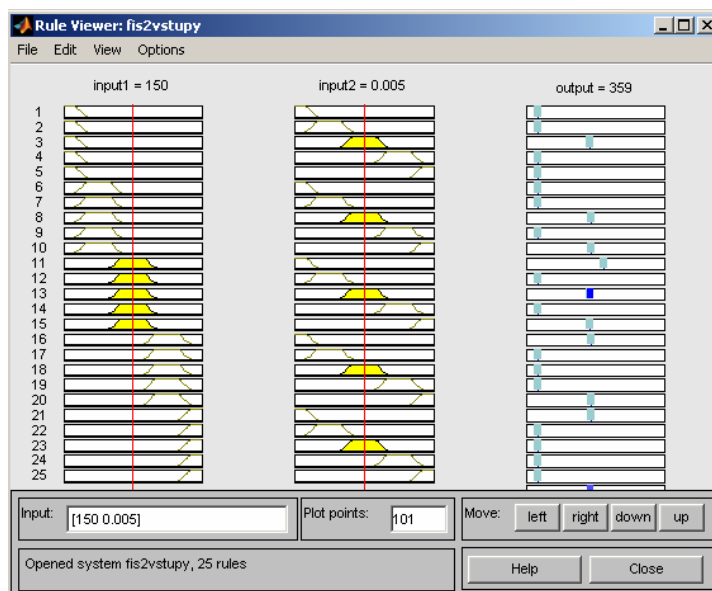


Obr. 4.13 Funkcie príslušnosti fuzzy modelu T-S



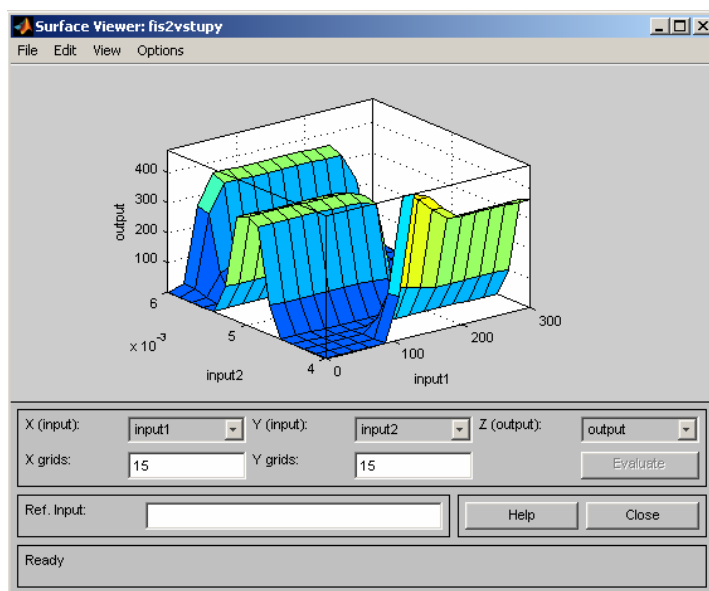
Obr. 4.14 Porovnanie výstupu z Anfis štruktúry

Výstupné údaje z Anfis štruktúry boli porovnané s dátami, ktoré sú určené na tréningovanie (obr. 4.14). V *rule vieweri* (obr. 4.15) sú zobrazené všetky pravidlá, tvary funkcií príslušnosti vstupov a výstupov a ich inferencie. Keďže máme dva vstupy a pre každý vstup uvažujeme 5 funkcií príslušnosti, počet pravidiel bude  $5 \times 5 = 25$ .



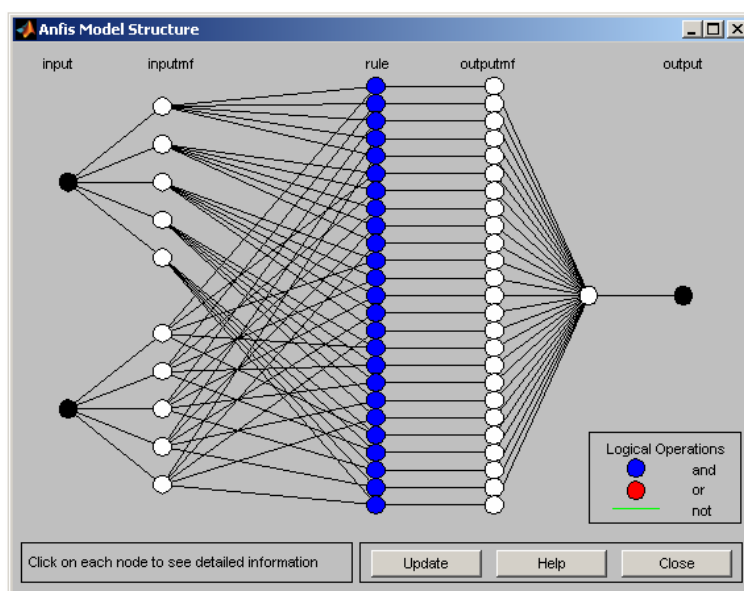
Obr. 4.15 Grafické zobrazenie procesu inferencie

*Surface viewer* vytvára trojdimenzionálny povrch vstupných a výstupných premenných, vytvára plochy ohraničujúce priestor generovaných akčných zásahov (obr.4.16).



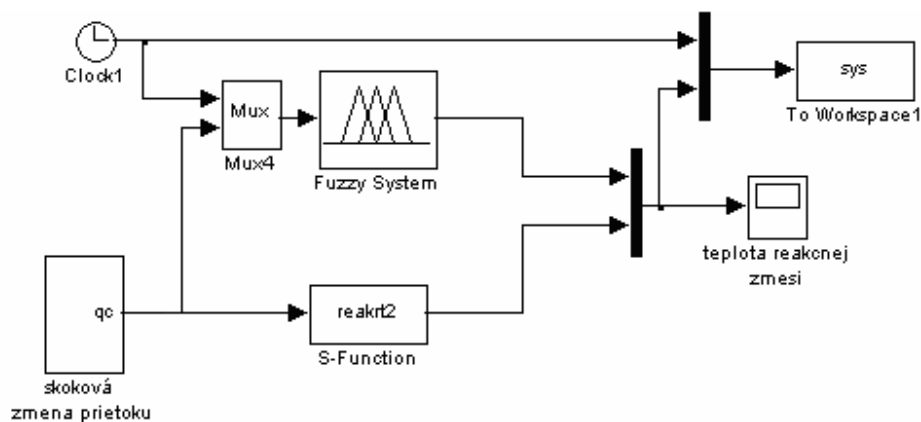
Obr. 4.16. Povrch vstupných a výstupných premenných

Výsledná Anfis štruktúra navrhnutého fuzzy modelu pre dva vstupy a jeden výstup je znázornená na obr. 4.17.



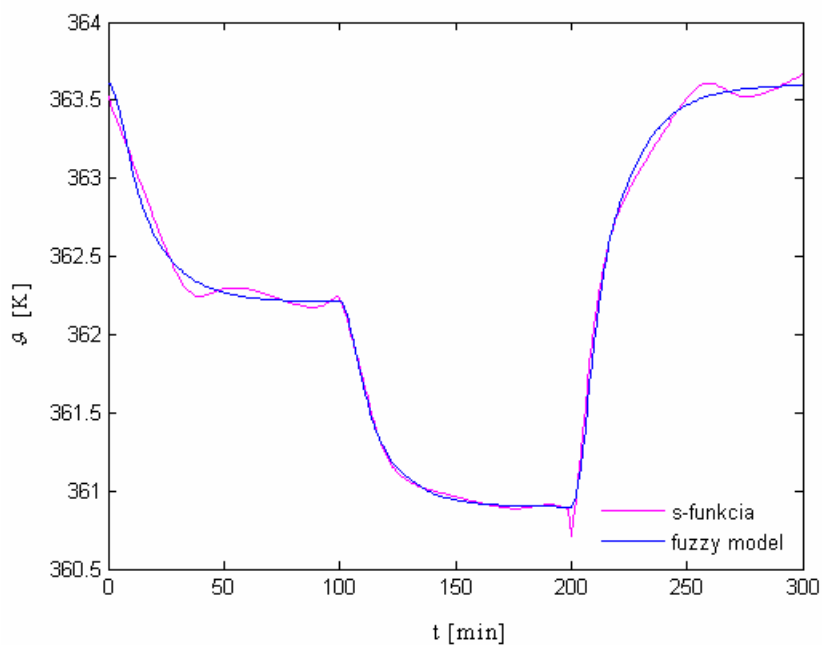
Obr. 4.17 Výsledná Anfis štruktúra

Na overenie správnosti návrhu fuzzy systému bola použitá nasledujúca bloková schéma.



Obr. 4.18 Bloková schéma na simuláciu s fuzzy systémom

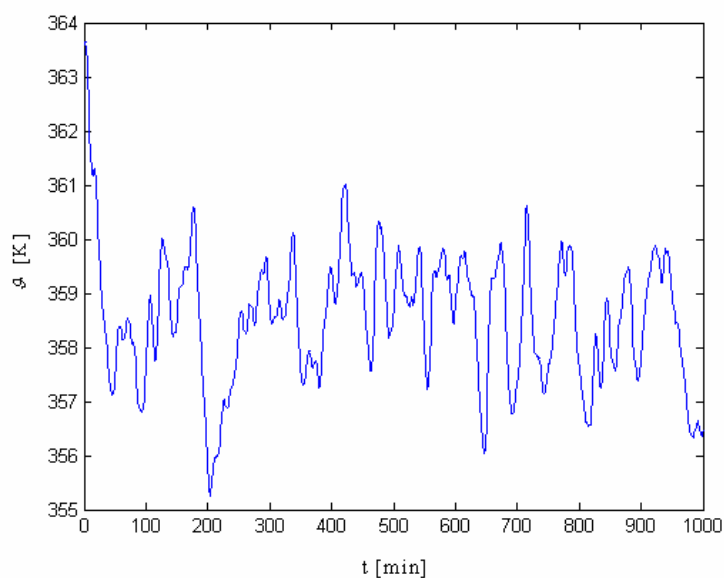
Pomocou grafického porovnania (obr.4.19) môžeme usúdiť, že rozdiel medzi prechodovou charakteristikou simulovanou s-funkciou a fuzzy systémom (*fis2vstupy.fis*) je minimálny. Program *fs2vstupy.m* na vytvorenie fuzzy systému je uvedený v prílohe.



Obr. 4.19 Porovnanie prechodových charakteristik

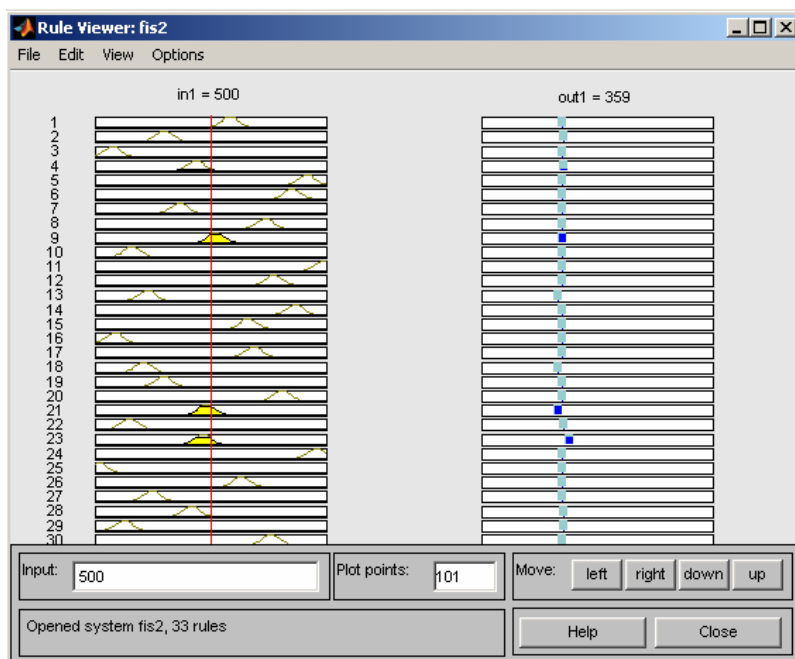
### 4.3 Fuzzy model s náhodným signálom na vstupe

Cieľom práce je prediktívne riadenie s fuzzy modelom. Aby fuzzy model lepšie vystihoval správanie sa reaktora v celom rozsahu zmeny vstupnej veličiny, bol fuzzy model navrhnutý pre náhodnú skokovú zmenu prietoku chladiaceho média v intervale od 0.004 do 0,016 m<sup>3</sup>min<sup>-1</sup> s periódou vzorkovania  $T_s = 1$  min. Priebeh teploty reakčnej zmesi od skokovej zmeny chladiaceho média je znázornený na obr. 4.20. Pri projektovaní fuzzy modelu je vstupnou veličinou čas a výstupnou veličinou sú namerané teploty reakčnej zmesi.



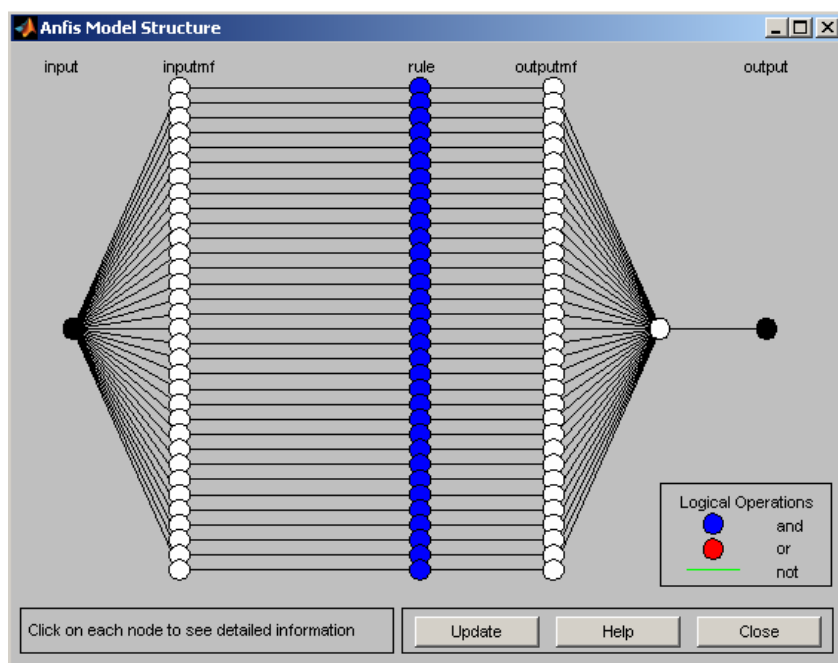
Obr. 4.20 Priebeh teploty reakčnej zmesi

Na inicializáciu fuzzy systému použijeme príkaz `genfis2`. Po inicializácii nasleduje tréning siete pomocou príkazu `Anfis`. `Anfis` trénuje funkcie príslušnosti parametrov FIS na modelovanie vstupných a výstupných dát. V *rule viewer* (obr. 4.21) sú zobrazené všetky pravidlá, tvary funkcií príslušnosti vstupov, výstupov a ich inferencie.



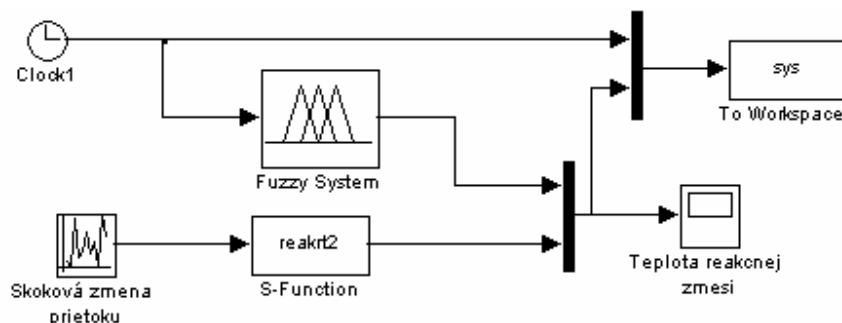
Obr. 4.21 Grafické zobrazenie procesu inferencie

Pri návrhu použijeme 33 funkcií príslušnosti typu *gaussmf*. Výsledná Anfis štruktúra navrhnutého fuzzy modelu je znázornený na obr. 4.22.



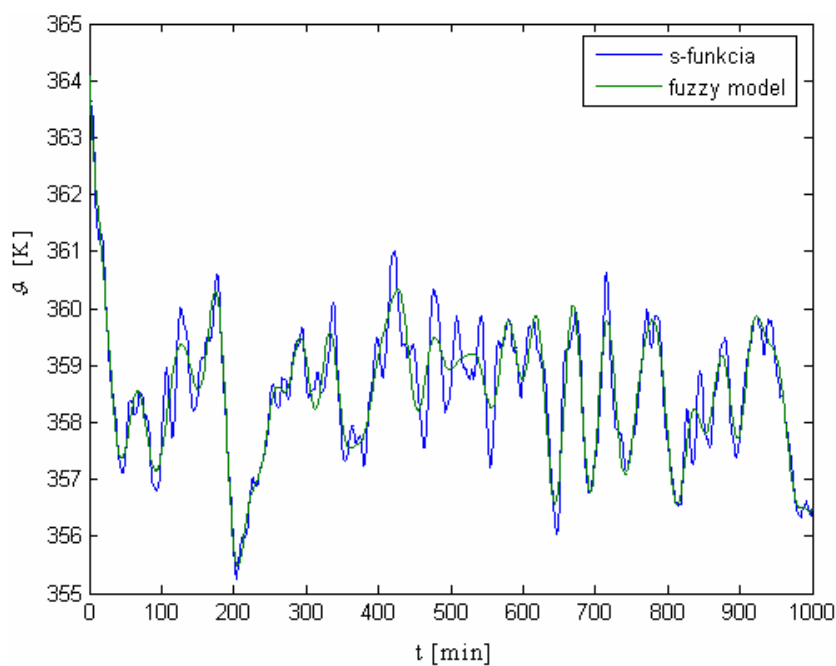
Obr. 4.22 Výsledná Anfis štruktúra

Na overenie správnosti návrhu fuzzy systému so skokovým signálom na vstupe bola použitá nasledujúca bloková schéma:



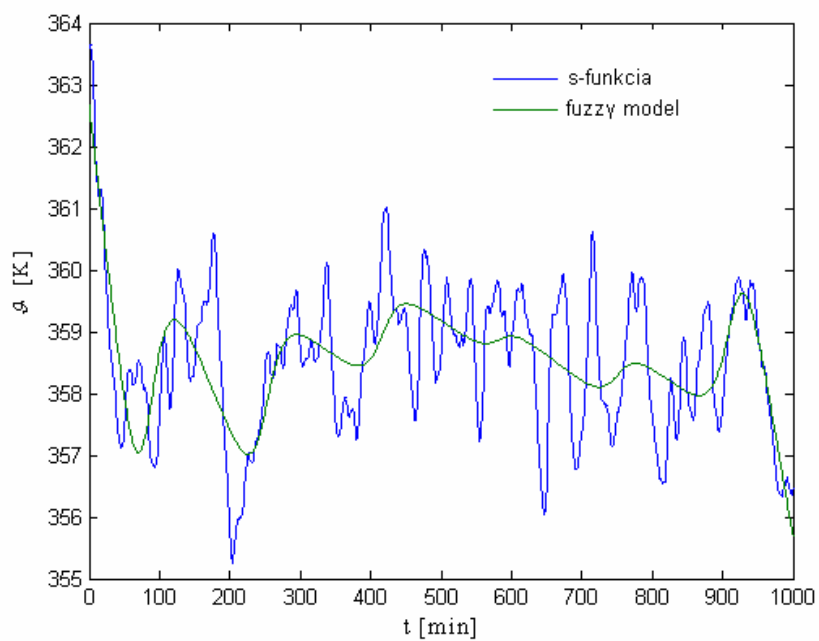
Obr. 4.23 Bloková schéma simulácie

Obr. 4.24 poukazuje na správnosť návrhu fuzzy modelu (*fis2.fis*) na základe nameraných údajov.



Obr. 4.24 Priebeh teploty reakčnej zmesi

Obr. 4.25 znázorňuje fuzzy model (*fis1.fis*) s náhodným vstupným signálom pri inicializácii príkazom *genfis1*. Postup pri návrhu bol podobný ako v sekcii 4.1.1.



Obr. 4.25 Priebeh teploty reakčnej zmesi

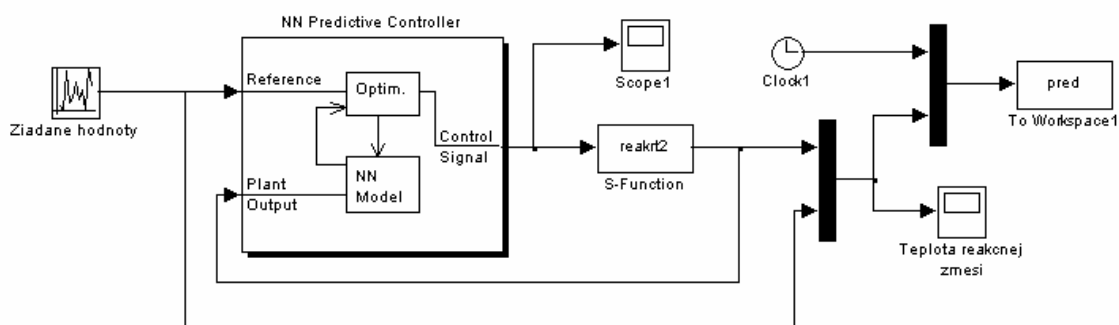
Fuzzy modely navrhnuté s náhodným vstupným signálom boli použité pri prediktívnom riadení.

## 5 Návrh prediktívneho riadenia

Prediktívne riadenie patrí medzi metódy riadenia, ktoré matematický model systému používajú na predikciu budúceho riadeného výstupu systému. Našou úlohou je navrhnúť prediktívne riadenie chemického reaktora.

### 5.1 Nastavenie prediktívneho regulátora

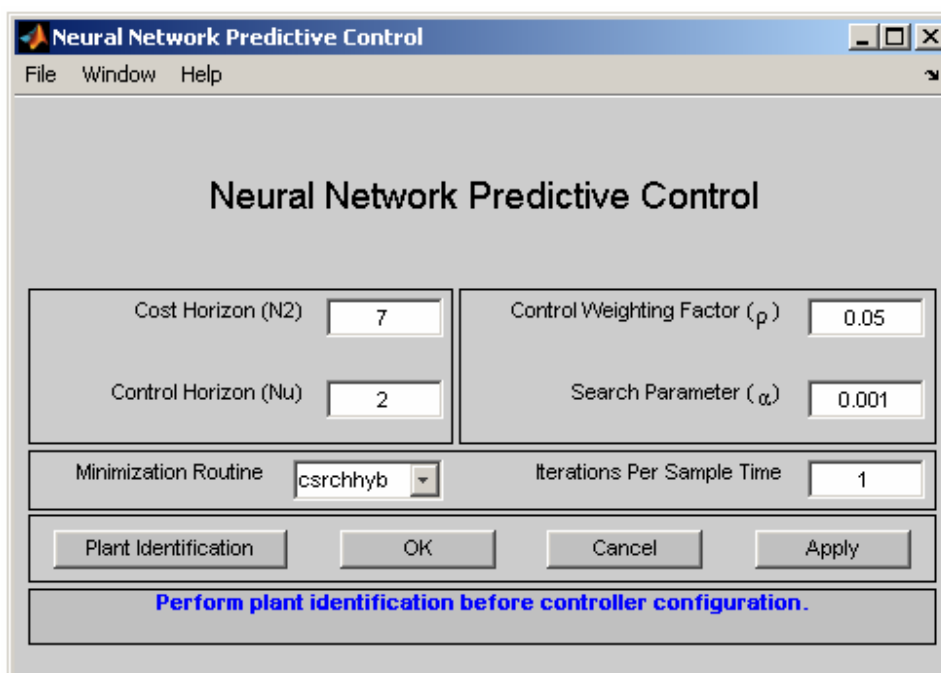
Na riadenie sme vytvorili model systému pomocou softvéru Matlab. Prediktívny regulátor na základe neurónovej siete je súčasťou toolboxu *Neural Network Blockset*. Bloková schéma používaná na simuláciu je znázornená na obr. 5.1.



Obr. 5.1 Bloková schéma prediktívneho riadenia

Prediktívny regulátor má dva vstupy, prvým vstupom je žiadaná hodnota a druhým vstupom je výstup z procesu. V bloku s-function je umiestnená s-funkcia, ktorá predstavuje riadený proces.

Prvým krokom pri riadení s prediktívnym regulátorom je správne nadefinovanie parametrov regulátora. Parametre nastavujeme v okne *Neural Network Predictive Control* (obr. 5.2).



Obr. 5.2 Konfigurácia prediktívneho regulátora

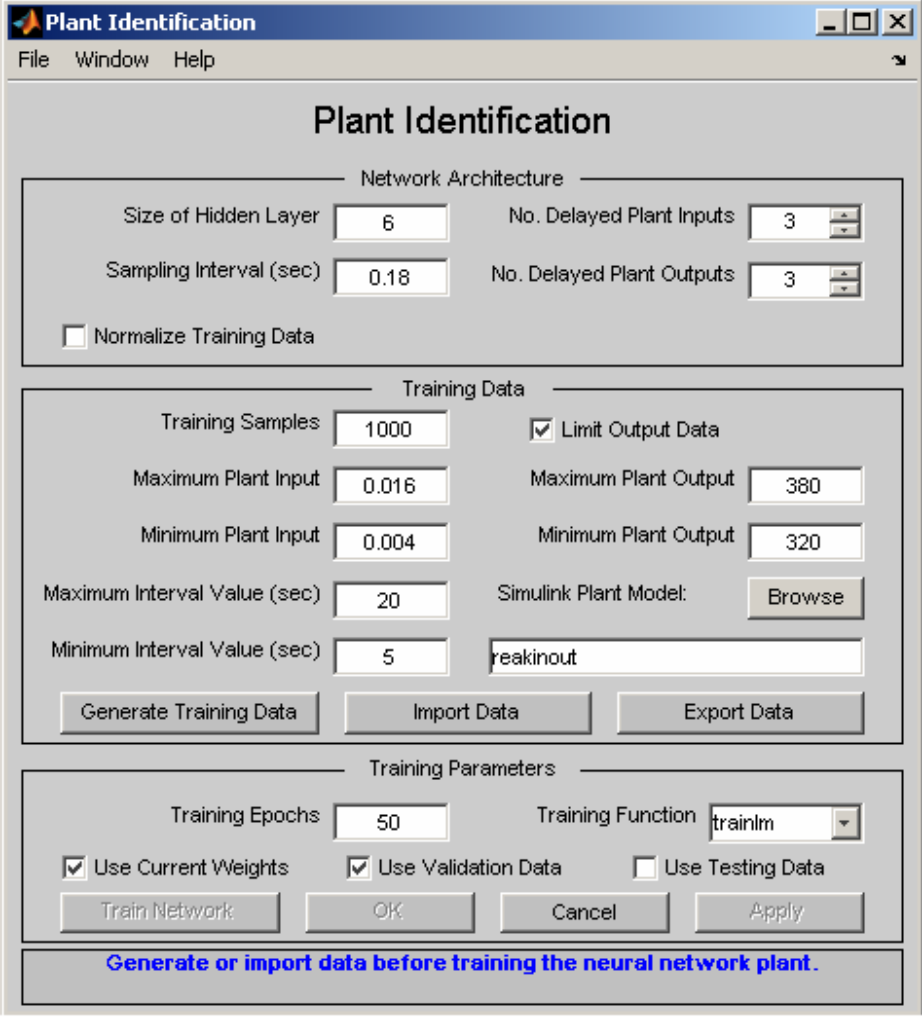
$N_2$  je účelový horizont, ktorý vyjadruje číselnú hodnotu kroku, nad ktorým je minimalizovaná predikčná chyba.  $N_u$ , riadiaci horizont, vyjadruje číselnú hodnotu kroku, nad ktorým je minimalizované riadenie.  $N_u$  musíme zvoliť tak, aby jeho hodnota bola menšia než hodnota  $N_2$ . Váhový faktor riadenia násobí sumu odchýlok riadenia v účelovej funkcii. Parameter  $\alpha$  určuje kedy je lineárne vyhľadávanie zastavené. Minimalizačnou metódou bola zvolená metóda *csrchhyb*, ktorá používa techniku kombinácie rozptylu a kubickej interpolácie. Úlohou minimalizačnej funkcie je hľadanie minima účelovej funkcie pri riadení.

Po nastavení parametrov regulátora nasleduje identifikácia procesu.

## 5.2 Identifikácia procesu

Nato, aby sme mohli úspešne spustiť regulátor, potrebujeme navrhnuť model procesu v tvare fuzzy modelu. Použité fuzzy modely boli navrhnuté v kapitole 4 a použijeme ich na trénovanie neurónovej siete. Parametre pre identifikáciu sa nastavujú

v ponuke *Plant Identification*, umiestnenej v ľavej dolnej časti okna prediktívneho regulátora (obr. 5.3).



**Plant Identification**

File Window Help

**Network Architecture**

Size of Hidden Layer: 6

Sampling Interval (sec): 0.18

No. Delayed Plant Inputs: 3

No. Delayed Plant Outputs: 3

☐ Normalize Training Data

**Training Data**

Training Samples: 1000

Maximum Plant Input: 0.016

Minimum Plant Input: 0.004

Maximum Interval Value (sec): 20

Minimum Interval Value (sec): 5

☒ Limit Output Data

Maximum Plant Output: 380

Minimum Plant Output: 320

Simulink Plant Model: Browse

reakinout

Generate Training Data Import Data Export Data

**Training Parameters**

Training Epochs: 50

Training Function: trainlm

☒ Use Current Weights ☒ Use Validation Data ☐ Use Testing Data

Train Network OK Cancel Apply

**Generate or import data before training the neural network plant.**

Obr. 5.3 Identifikácia procesu

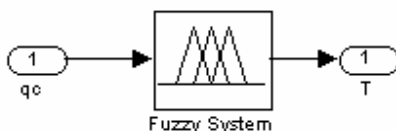
Okno identifikácie je rozdelené na tri časti.

V prvej časti sa nastavuje sieťová štruktúra (*Network Architecture*). V bloku *Size of Hidden Layer* sa nastavuje počet neurónov v prvej vrstve neurónovej siete. *Sampling interval* predstavuje interval, v ktorom program zbiera dáta z modelu procesu. V našom prípade je to fuzzy model. *Delayed Plant Inputs* a *Outputs* sú oneskorené vstupy a výstupy.

V ďalšej časti sa trénujú údaje z fuzzy modelu. *Training samples* určuje počet referenčných bodov vygenerovaných pre tréning a overenie správnosti. Maximálne

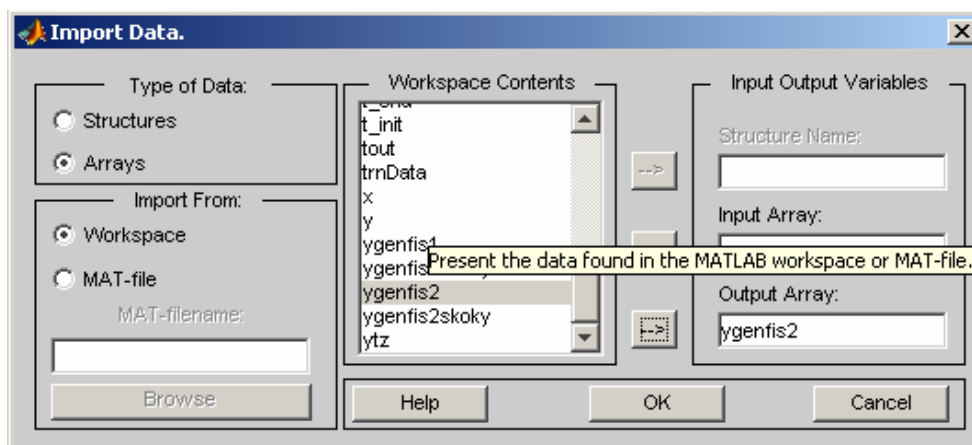
a minimálne hodnoty vstupov a výstupov zadávame v nasledujúcich štyroch blokoch : *Maximum Plant Input/Output*, *Minimum Plant Input/Output*. V blokoch *Maximum/Minimum Interval value* nastavujeme čas hornej/dolnej hranice skokovej zmeny vstupnej veličiny.

Trénovanie údajov môže byť realizované dvoma spôsobmi. Prvým je spôsob, kedy generujeme trérovacie údaje pomocou blokovej schémy (obr. 5.4). Vtedy názov blokovej schémy zadáme do poľa *Simulink Plant Model*.



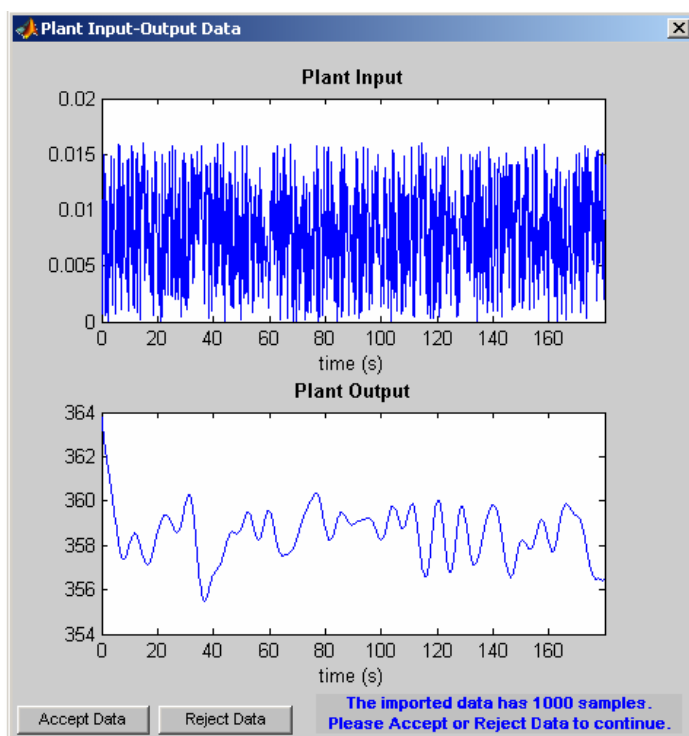
Obr. 5.4 Bloková schéma generovania údajov

Druhým spôsobom je importovanie údajov z prechodovej charakteristiky reaktora. Vstupnou veličinou je prietok chladiaceho média, ktorú zadávame v okne *Import Data* (obr.5.5). Výstupnou veličinou je teplota reakčnej zmesi, ktorá je výstupom z fuzzy modelu navrhnutého pomocou metódy *genfis1*.



Obr. 5.5 Importovanie údajov

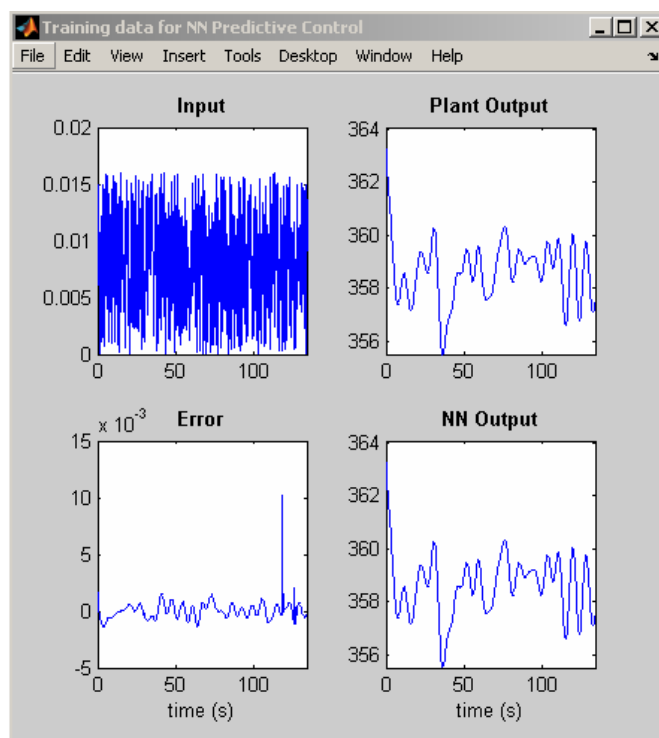
Po výbere vstupu a výstupu automaticky vygeneruje trérovacie dáta (obr. 5.6)



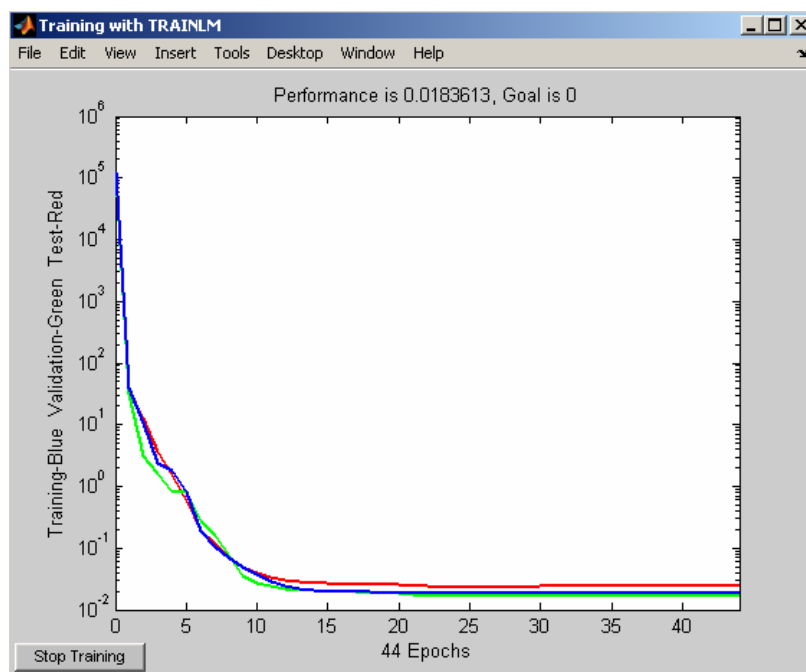
Obr. 5.6 Vstupno-výstupné údaje procesu

Tretou časťou je tréovanie parametrov. *Training Epochs* vyžaduje počet iterácií potrebných na tréovanie procesu. V rozbaľovacom poli *Training Function* môžeme vybrať jednu z trinástich typov funkcií tréovania, napríklad *trainlm*. *Trainlm* predstavuje sieťovú tréovaciu funkciu, ktorá aktualizuje váhu na základe *Levenberg - Marquardt* optimalizácie. *Train Network* slúži na spustenie tréovania parametrov. Po ukončení procesu získame výsledok testovania, validácie a tréovania. Z obr. 5.7 vyplýva, že rozdiel medzi výstupom z fuzzy modelu a z natréovanej neurónovej siete je minimálny.

Porovnanie tréovania, validácie a testovania pomocou funkcie *trainlm* je na obr. 5.8. Výsledky tréovania aplikujeme na prediktívny regulátor, ktorý je následne pripravený na riadenie reaktora.



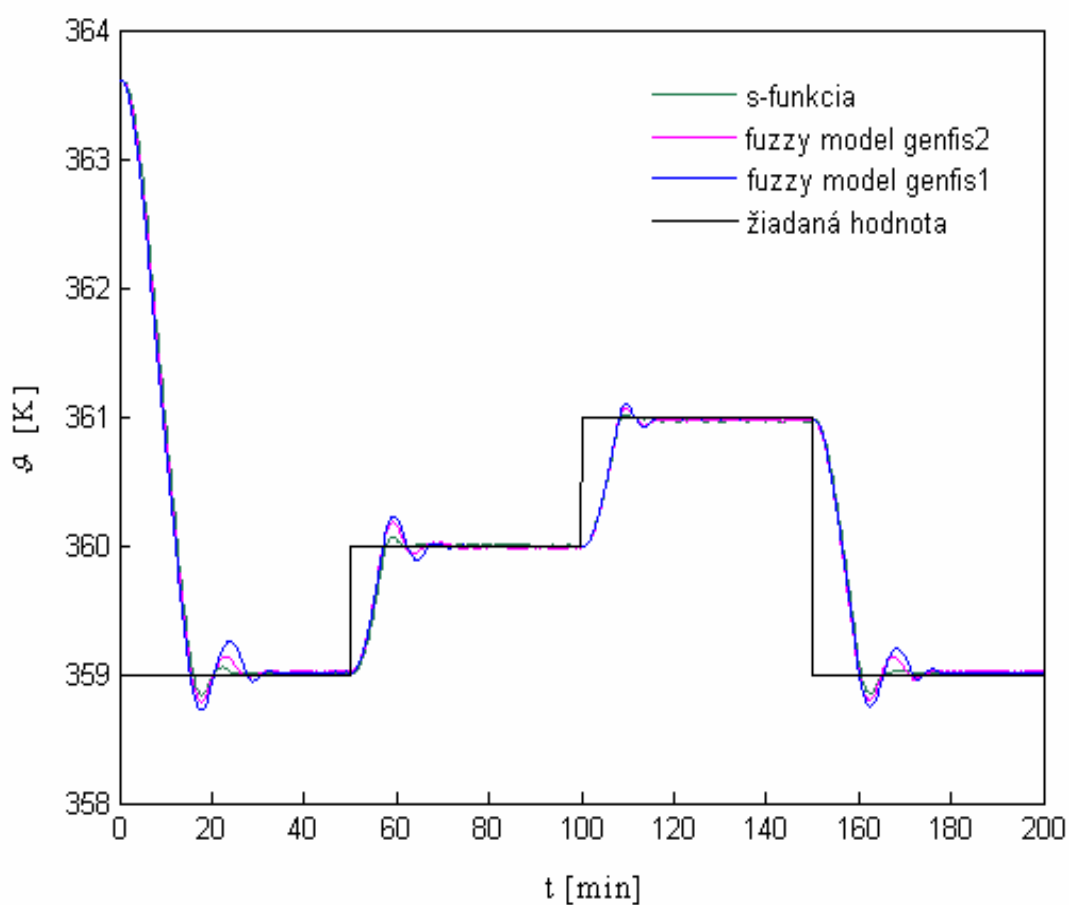
Obr. 5.7 Trénovanie neurónovej siete



Obr. 5.8 Porovnanie trénovania, validácie a testovania

### 5.3 Prediktívne riadenie chemického reaktora

Na riadenie chemického reaktora sme použili prediktívne riadenie s fuzzy modelom. Z navrhnutých modelov boli vybrané dva fuzzy modely s rôznymi spôsobmi inicializácie. Prediktívne riadenie chemického reaktora pomocou fuzzy modelov bolo porovnané s prediktívnym riadením s modelom reaktora v tvare s-funkcie, ktorá predstavovala zároveň riadený proces. Výsledky riadenia sú znázornené v obr. 5.9. Z obrázkov vyplýva, že s fuzzy modelom je možné dosiahnuť riadenie porovnateľné s riadením pomocou s-funkcie.



Obr. 5.9 Porovnanie prediktívneho riadenia chemického reaktora  
fuzzy modelom a s-funkciou

## Záver

Cieľom práce bolo navrhnuť prediktívne riadenie s fuzzy modelom. Fuzzy model bolo potrebné navrhnuť tak, aby opisoval správanie sa riadeného procesu.

Predmetom riadenia bol prietokový chemický reaktor s dvoma exotermickými paralelnými reakciami. Nelineárny model reaktora bol použitý na získanie údajov potrebných pri návrhu fuzzy modelov. Jednou z možností ako získať fuzzy model je metóda Anfis. Podstatou tejto metódy je neuro – adaptívny proces učenia, na základe ktorého možno z nameraných údajov odvodiť parametre funkcií príslušnosti a extrahovať bázu pravidiel. Tieto pravidlá boli potrebné na návrh fuzzy modelu. Na inicializáciu fuzzy systému boli vykúšané príkazy genfis1 a genfis2. Na ilustráciu postupu návrhu fuzzy modelu sú použité vstupné údaje namerané pri troch skokových zmenách na vstupe. Aby fuzzy model lepšie vystihoval správanie sa reaktora v celom rozsahu zmeny vstupnej veličiny, boli navrhnuté fuzzy modely pre náhodnú skokovú zmenu prietoku chladiaceho média. Takto získané fuzzy modely boli aplikované pri prediktívnom riadení.

Pri prediktívnom riadení je možné uvažovať obmedzenia na vstupné a výstupné veličiny priamo pri návrhu regulátora. Obmedzenia potrebné na nastavenie regulátora sú výsledkom analýzy chemického reaktora.

Na riadenie chemického reaktora bolo použité prediktívne riadenie s fuzzy modelom. Na riadenie boli vybraté dva fuzzy modely s náhodným vstupným signálom a s rôznymi postupmi inicializácie. Prediktívne riadenie chemického reaktora pomocou fuzzy modelov bolo porovnané s prediktívnym riadením s modelom reaktora v tvare s-funkcie, ktorá predstavovala zároveň riadený proces. Z porovnania výsledkov riadenia vyplýva, že navrhnuté fuzzy modely sú použiteľné pre riadenie chemického reaktora. Pri riadení s fuzzy modelom je možné dosiahnuť riadenie porovnateľné s riadením pomocou s-funkcie.

## Literatúra

- [1] Babjak, J. 2003. *Ako pracujú fuzzy systémy*. [20-02-2007]. Dostupné na: [<http://www.root.cz/clanky/ako-pracuju-fuzzy-systemy/>](http://www.root.cz/clanky/ako-pracuju-fuzzy-systemy/)
- [2] Babjak, J. 2003. *Hybridná inteligencia*. [10-04-2007] Dostupné na: [<http://www.root.cz/clanky/hybridna-inteligencia/>](http://www.root.cz/clanky/hybridna-inteligencia/)
- [3] Bakošová, M. *Prietokový chemický reaktor s dokonalým miešaním reakčnej zmesi*. Dostupné na : [<www.kirp.chtf.stuba.sk/~bakosova/wwwDP/p9.pdf>](http://www.kirp.chtf.stuba.sk/~bakosova/wwwDP/p9.pdf)
- [4] Bakošová, M., Vasičkaninová, A., Karšaiová, M., and Ondrovičová, M. 2004. *PID controller tuning for cascade control of a chemical reactor*. In: Proc. 6. International Scientific-Technical Conf. Process Control 2004. Kouty nad Desnou. University of Pardubice, p. 63. ISBN 80-7194-662-1.
- [5] B. W. Bequette 2002. *Process Control: Modeling, Design and Simulation*. Prentice Hall, 2002 ISBN 0-13-353640-8  
Dostupné na:< <http://search.safaribooksonline.com/0133536408>>
- [6] Demuth, H., Beale, M., Hogan, M. *Neural Network Toolbox 5 User's Guide*, 1992-2007 by The Math Works, Inc. p. 222-231
- [7] *Fuzzy logika*. [12-02-2007] Dostupné na:<<http://marcelm.szm.sk>>
- [8] Jura, P. 2003. *Základy fuzzy logiky pro řízení a modelování*. Vutium, 2003. ISBN 80-214-2261-0
- [9] Mikleš, J., Fikar, M. 2004. *Modelovanie, identifikácia a riadenie procesov II*, STU Bratislava, 2004. 208-232 s.
- [10] Modrlák, O. 2004. *Fuzzy řízení a regulace*. Technická Univerzita v Liberci 2004
- [11] Novák, V. 2000. *Základy fuzzy modelování*. Praha: Ben, 2000. 176 s. ISBN 80-7300-009-1
- [12] Petriščák, M. *Základy fuzzy logiky a jej aplikácie v Expertných systémoch*. Fakulta informatiky a informačných technológií. Bratislava. [20-02-2007]

- [13] Šoltys, J. 1996. *Použitie fuzzy neurónovej siete typu NARA na riadenie vozíka s balancujúcou tyčou*. Laboratorium umelej inteligencie [12-02-2007]. Dostupné na: [www.ai-cit.sk/source/publications/thesis/master\\_thesis/1996/soltys/html/all.html](http://www.ai-cit.sk/source/publications/thesis/master_thesis/1996/soltys/html/all.html)
- [14] Soloway, D., Haley, J. P. *Neural generalized predictive control*. Hampton: NASA Langley Research Center. p. 17.
- [15] Olej, V., Petr, P. *Model predikcie záverečnej ceny indexového fondu na báze fuzzy inferenčného systému Takagi – Sugeno*. Pardubice: Ústav systémového inžinýrství a informatiky
- [16] Vysoký, P. 1996. *Fuzzy řízení*, Praha: ČVUT, 1996. 33-37, 87-90 s. ISBN 80-01-01429-8

## Príloha

### A: Program na návrh fuzzy modelov fsngenfis1.fis a fsngenfis2.fis

```

open_system('reaktor2');
sim('reaktor2',300);
bdclose('reaktor2')
%%% návrh fuzzy systemu fsngenfis1
y=data(:,1);q=data(:,2);t=data(:,3);
trnData=[t y];

numMFs = 7;
mfType = 'trimf';
epoch_n = 1;
in_fisat = genfis1(trnData,numMFs,mfType);
writefis(in_fisat,'identif');
a=readfis('identif');
plotmf(a,'input',1)
out_fisat = anfis(trnData,in_fisat,3);
writefis(out_fisat,'fsngenfis1');
ew=evalfis(t,out_fisat);
figure
plot(t,y,t,ew),xlabel('t'), ylabel('y')
legend('Training Data','ANFIS Output')
ruleview('fsngenfis1')
surfview('fsngenfis1')
fisat=readfis('fsngenfis1.fis');
open_system('fslvstupskok');
sim('fslvstupskok',300);

%%% návrh fuzzy systemu fsngenfis2
x=[t];
out_fisat =genfis2(x,y,0.1 );
writefis(out_fisat,'fsngenfis2');
%type gfis1.fis
[ep,irr,orr,arr]=evalfis(x,out_fisat);
% save ew1.dat ew -ascii
figure
plot(t,y,t,ep);
legend('Training Data','genfis2 Output');pause
e=ep-y;plot(t,e);legend('chyba');
ruleview('fsngenfis2')
surfview('fsngenfis2')
fisat=readfis('fsngenfis2');
type fsngenfis2.fis
open_system('fslvstupskok');
sim('fslvstupskok',300);

```

## B: Program pre návrh fuzzy modelu pre viac vstupných premenných

```
open_system('reaktor2');
sim('reaktor2',300);
bdclose('reaktor2')
%% návrh fuzzy systému fis2vstup.fis
y=data(:,1);q=data(:,2);t=data(:,3);
x=[t q];
trnData = [x y];
numMFs =5;
mfType = 'gauss2mf';
epoch_n =1;
in_fismat = genfis1(trnData,numMFs,mfType);
writefis(in_fismat, 'fism1');
a=readfis('fism1');
figure
    plotmf(a,'input', 1),pause
    out_fismat = anfis(trnData,in_fismat,3);
    writefis(out_fismat, 'fis2vstup');
    a=readfis('fis2vstup');
    figure
        gensurf(a),title('povrch vystupnej fis matice, input1=t, outpu=y')
    ytz=evalfis(x,out_fismat)
    figure
    plot(t,y,t,ytz);
    ruleview('fis2vstup')
    surfview('fis2vstup')
    fismat=readfis('fis2vstup');
    type fis2vstup.fis
open_system('fs2vstup');
sim('fs2vstup',300);
figure
plot(sys(:,1),sys(:,2),'m',sys(:,1),sys(:,3))
legend('s-funkcia','fuzzy model')
```

## C: Štruktúra fuzzy inferenčného systému fis1.fis

```

[System]
Name='fis1'
Type='sugeno'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=7
AndMethod='prod'
OrMethod='max'
ImpMethod='prod'
AggMethod='max'
DefuzzMethod='wtaver'
[Input1]
Name='input1'
Range=[0 1000]
NumMFs=7
MF1='in1mf1':'gauss2mf',[28.3107266762673 -50 28.3257465128799
50.0069040943852]
MF2='in1mf2':'gauss2mf',[28.3188486905144 116.663484992173
28.3074930130068 216.664513137207]
MF3='in1mf3':'gauss2mf',[28.308776294577 283.333261601732
28.3083103044292 383.33076685323]
MF4='in1mf4':'gauss2mf',[28.3133641883646 449.997331426644
28.3099448448626 549.999399393458]
MF5='in1mf5':'gauss2mf',[28.3111657541717 616.666239482891
28.3093104483937 716.666297757372]
MF6='in1mf6':'gauss2mf',[28.3105924811801 783.333641546755
28.3099002284729 883.333291194483]
MF7='in1mf7':'gauss2mf',[28.3116981543795 950.000101772902
28.3107266762673 1050]
[Output1]
Name='output'
Range=[355.262560829816 363.651757826309]
NumMFs=7
MF1='out1mf1':'linear',[-0.0987859091891636 362.660022639765]
MF2='out1mf2':'linear',[-0.0276186262293238 362.900011268389]
MF3='out1mf3':'linear',[-0.00785486714981831 361.360508732167]
MF4='out1mf4':'linear',[-0.00747007394427523 362.913649462795]
MF5='out1mf5':'linear',[-0.00845989987546402 364.124676704438]
MF6='out1mf6':'linear',[-0.00787142742924726 364.701284508065]
MF7='out1mf7':'linear',[-0.066977854922596 422.651081829615]
[Rules]
1, 1 (1) : 1
2, 2 (1) : 1
3, 3 (1) : 1
4, 4 (1) : 1
5, 5 (1) : 1
6, 6 (1) : 1
7, 7 (1) : 1

```

## D: Štruktúra fuzzy inferenčného systému fis2.fis

```

[System]
Name='fis2'
Type='sugeno'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=33
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='wtaver'
[Input1]
Name='in1'
Range=[0 1000]
NumMFs=33
MF1='in1mf1':'gaussmf',[35.3553390593274 585]
MF2='in1mf2':'gaussmf',[35.3553390593274 300]
MF3='in1mf3':'gaussmf',[35.3553390593274 76]
MF4='in1mf4':'gaussmf',[35.3553390593274 438]
MF5='in1mf5':'gaussmf',[35.3553390593274 916]
MF6='in1mf6':'gaussmf',[35.3553390593274 853]
MF7='in1mf7':'gaussmf',[35.3553390593274 368]
MF8='in1mf8':'gaussmf',[35.3553390593274 737]
MF9='in1mf9':'gaussmf',[35.3553390593274 523]
MF10='in1mf10':'gaussmf',[35.3553390593274 161]
MF11='in1mf11':'gaussmf',[35.3553390593274 987]
MF12='in1mf12':'gaussmf',[35.3553390593274 774]
MF13='in1mf13':'gaussmf',[35.3553390593274 227]
MF14='in1mf14':'gaussmf',[35.3553390593274 868]
MF15='in1mf15':'gaussmf',[35.3553390593274 664]
MF16='in1mf16':'gaussmf',[35.3553390593274 86]
MF17='in1mf17':'gaussmf',[35.3553390593274 688]
MF18='in1mf18':'gaussmf',[35.3553390593274 211]
MF19='in1mf19':'gaussmf',[35.3553390593274 296]
MF20='in1mf20':'gaussmf',[35.3553390593274 809]
MF21='in1mf21':'gaussmf',[35.3553390593274 482]
MF22='in1mf22':'gaussmf',[35.3553390593274 150]
MF23='in1mf23':'gaussmf',[35.3553390593274 467]
MF24='in1mf24':'gaussmf',[35.3553390593274 957]
MF25='in1mf25':'gaussmf',[35.3553390593274 16]
MF26='in1mf26':'gaussmf',[35.3553390593274 632]
MF27='in1mf27':'gaussmf',[35.3553390593274 248]
MF28='in1mf28':'gaussmf',[35.3553390593274 418]
MF29='in1mf29':'gaussmf',[35.3553390593274 127]
MF30='in1mf30':'gaussmf',[35.3553390593274 761]
MF31='in1mf31':'gaussmf',[35.3553390593274 644]
MF32='in1mf32':'gaussmf',[35.3553390593274 968]
MF33='in1mf33':'gaussmf',[35.3553390593274 557]

```

```

[Output1]
Name='out1'
Range=[355.262560829816 363.651757826309]
NumMFs=33
MF1='outlmf1':'linear',[137557.974857401 -89514455.8508002]
MF2='outlmf2':'linear',[80983.2117008216 0]
MF3='outlmf3':'linear',[19535.2543784758 5259548.1070362]
MF4='outlmf4':'linear',[2067468.98071312 -884303717.371969]
MF5='outlmf5':'linear',[6704.81608509662 -5777362.38254004]
MF6='outlmf6':'linear',[-19625.989138534 14968420.1776089]
MF7='outlmf7':'linear',[20543.7912922597 -6644644.0441383]
MF8='outlmf8':'linear',[41307.0000445382 -27392563.585287]
MF9='outlmf9':'linear',[396649.60980003 -212959488.607999]
MF10='outlmf10':'linear',[-110394.427733271 0]
MF11='outlmf11':'linear',[11468.5250550199 -14105041.2175053]
MF12='outlmf12':'linear',[-5365.7001252142 0]
MF13='outlmf13':'linear',[-2328219.21530098 632379275.417939]
MF14='outlmf14':'linear',[-8412.94833805371 9396144.6132287]
MF15='outlmf15':'linear',[10706.0291818272 0]
MF16='outlmf16':'linear',[47704.8898500967 -9873123.49601151]
MF17='outlmf17':'linear',[41325.0855601575 -31042276.6896261]
MF18='outlmf18':'linear',[-1105140.12956074 45639217.4126791]
MF19='outlmf19':'linear',[-200639.63368028 38357219.7726618]
MF20='outlmf20':'linear',[-10967.5615642559 9142983.23637993]
MF21='outlmf21':'linear',[3370150.6119932 -2170244170.7418]
MF22='outlmf22':'linear',[321453.46258722 -47780947.6483592]
MF23='outlmf23':'linear',[5186038.02940028 -1970462413.17302]
MF24='outlmf24':'linear',[130095.993477683 -95643982.4392079]
MF25='outlmf25':'linear',[4.91201057960839 1044.9468243044]
MF26='outlmf26':'linear',[130584.761876616 -71837386.4874462]
MF27='outlmf27':'linear',[-634538.506070222 241684290.793321]
MF28='outlmf28':'linear',[554908.813088013 -142513292.913626]
MF29='outlmf29':'linear',[120228.69039248 -2592268.4231229]
MF30='outlmf30':'linear',[72109.8751829508 -54808592.5991328]
MF31='outlmf31':'linear',[-25465.4731782539 0]
MF32='outlmf32':'linear',[167577.516736786 -188716376.132543]
MF33='outlmf33':'linear',[288765.132225811 -161969559.637104]
[Rules]
1, 1 (1) : 1
2, 2 (1) : 1
3, 3 (1) : 1
4, 4 (1) : 1
5, 5 (1) : 1
6, 6 (1) : 1
7, 7 (1) : 1
8, 8 (1) : 1
9, 9 (1) : 1
10, 10 (1) : 1
11, 11 (1) : 1
12, 12 (1) : 1
13, 13 (1) : 1
14, 14 (1) : 1
15, 15 (1) : 1
16, 16 (1) : 1
17, 17 (1) : 1
18, 18 (1) : 1
19, 19 (1) : 1
20, 20 (1) : 1
21, 21 (1) : 1
22, 22 (1) : 1
23, 23 (1) : 1
24, 24 (1) : 1
25, 25 (1) : 1
26, 26 (1) : 1
27, 27 (1) : 1
28, 28 (1) : 1
29, 29 (1) : 1
30, 30 (1) : 1
31, 31 (1) : 1
32, 32 (1) : 1
33, 33 (1) : 1

```