



SLOVENSKÁ TECHNICKÁ UNIVERZITA

Fakulta chemickej a potravinárskej technológie

Katedra informatizácie a riadenia procesov

Radlinského 9, 812 37 Bratislava

## **Fuzzy modelovanie dynamiky chemického procesu v prostredí Matlab.**

Školiteľ: Ing. Anna Vasičkaninová

Vypracovala: Zuzana Borošová

Bratislava 2005



Ďakujem vedúcej  
bakalárskej práce Ing. Anne  
Vasičkaninovej za pomoc  
pri získavaní vedomostí z  
oblasti fuzzy modelovania a  
celkovo za vedenie, rady a  
pripomienky, ktoré mi  
poskytla pri vypracovaní  
bakalárskej práce.

<b>1. ÚVOD.....</b>	<b>4</b>
<b>2. TEORETICKÁ ČASŤ.....</b>	<b>5</b>
2.1. Fuzzy logika.....	5
2.2. Fuzzy množiny.....	5
2.3. Fuzzyfikácia.....	7
2.4. Defuzzyfikácia .....	8
2.5. Modelovanie procesov.....	8
2.6. Fuzzy modely .....	9
<b>3. PRAKTICKÁ ČASŤ .....</b>	<b>10</b>
3.1. Chemické reaktory.....	10
3.2. Fuzzy model chemického reaktora .....	11
3.3. Vytvorenie fuzzy modelu .....	12
3.3.1. FIS Editor – Inferenčný systém fuzzy .....	12
3.3.2. Membership Funktion Editor – Editor funkcií príslušnosti .....	15
3.3.3. Rule Editor Editor – Editor pravidiel.....	15
3.3.4. Rule Viewer – Grafické zobrazenie procesu inferencie.....	16
3.3.5. Surface Viewer – Plochy ohraničujúce priestor generovaných akčných zásahov .....	17
3.4. Anfis .....	17
<b>4. TVORBA FUZZY MODELOV.....</b>	<b>20</b>
4.1. Prvý príklad.....	20
4.2. Druhý príklad.....	23
4.3. Tretí príklad .....	26
<b>5. ZÁVER.....</b>	<b>29</b>
<b>6. LITERATÚRA.....</b>	<b>30</b>
<b>7. PRÍLOHY.....</b>	<b>31</b>

# 1. ÚVOD

Človek sa málokedy stretáva s presne definovanými problémami, ktorými sú napr. rôzne matematické problémy, ako napr. výpočet obsahu rôznych telies alebo vyriešenie rovnice s neznámymi premennými. Pri riešení takýchto - presne definovaných úloh je jednoduché využiť počítač. Čo ale s úlohami, ktoré pracujú s neurčitostami.

Existujú rôzne metódy vo fuzzy teórii, ktoré napriek nepresným, neurčitým informáciám, umožňujú odvodiť vyhovujúce, alebo aspoň prijateľné riešenie.

V posledných rokoch sa v praxi môžeme stretnúť v riadení a regulácii s prístupmi a princípmi, ktoré sú založené na vednej disciplíne označenej ako *SC – Soft Computing*. Medzi hlavných zástupcov SC zahŕňame *fuzzy logiku, neurónové siete a genetické algoritmy*.

Keďže som sa v mojej práci zaoberala fuzzy modelovaním dynamiky chemického procesu, základom bol pôvodný návrh modelu.

Návrh tvaru, druhu a počtu funkcií 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 FP sa pokiaľ možno používa čo najjednoduchší, kvôli zníženiu výpočtovej náročnosti pri ich vyhodnocovaní; zvyčajne sú to tzv. trojuholníkové alebo lichobežníkové FP. Zaoberala som sa porovnávaním použitých druhov, počtov funkcií príslušnosti pri jednotlivých modeloch.

## 2. TEORETICKÁ ČASŤ

### 2.1. Fuzzy logika

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. Veľmi populárne sú systémy na podporu rozhodovania, kde úloha fuzzy logiky spočíva vo výbere najvhodnejšej alternatívy.

Aplikačne dosahuje rozsah pôsobenia od spotrebných tovarov, až po priemyselné riadenie procesov. Jej princíp spočíva v rozšírení logických operátorov na fuzzy množiny.

### 2.2. Fuzzy množiny

Teória fuzzy množín spočíva v zavedení tzv. *stupňa príslušnosti* prvku k množine, ktorá môže nadobúdať hodnoty z intervalu  $\langle 0,1 \rangle$  na rozdiel od klasických teórií množín, kde každý prvok do množiny buď patrí alebo nepatrí.

V klasickej teórii množín je možné množinu popísať niekoľkými spôsobmi:

- výpočtom prvkov množiny  $M = \{x_1, x_2, x_3, x_4\}$ ,
- pravidlom, ktorému musia prvky vyhovovať  $m_M(x)$ ,
- *charakteristickou funkciou*  $m_M(x)$ , pre ktorú platí.

$$m_M(x) = \begin{cases} 1, & \text{ak } x \in M \\ 0, & \text{ak } x \notin M \end{cases}$$

*Ostrá množina* - Ak prvok v klasickej teórii množín do množiny buď patrí, alebo nepatrí, pretože jeho charakteristická funkcia nadobúda hodnoty 1 alebo 0.

*Neostrá (fuzzy) množina* - Pokiaľ charakteristická funkcia charakterizuje stupeň, s ktorým prvok do množiny patrí.

*Lingvistická premenná* – Je to premenná, ktorej hodnoty sú výrazmi nejakého jazyka. Hodnotu lingvistickej premennej môžeme interpretovať ako fuzzy – neostré množiny.

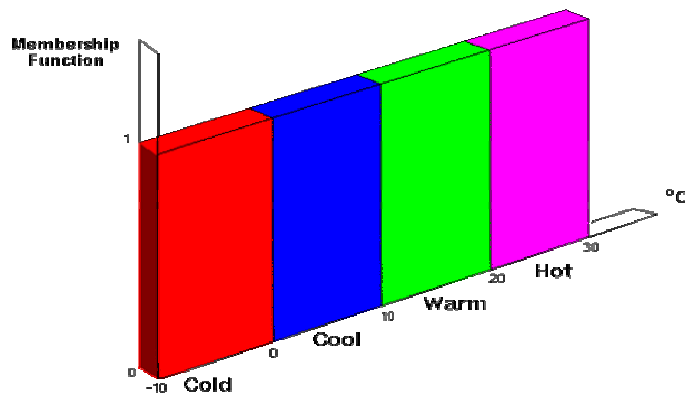
Rozsiahle uplatnenie fuzzy množín v regulačnej technike a v mnohých technických disciplínach je založené na dvoch základných princípoch:

- pomocou fuzzy množín môžeme aproximovať ľubovoľnú spojitú funkciu,
- pomocou fuzzy množín môžeme formalizovať znalosti, ktoré sú vo vágnej forme.

*Funkcia príslušnosti* je charakteristická funkcia  $\mu_S(x)$  pri neostrých množinách. Charakterizuje stupeň, s ktorým daný prvok patrí do príslušnej množiny a to od hodnoty 0, keď prvok do množiny určite nepatrí, až do hodnoty 1 kedy prvok do množiny určite patrí.

Návrh tvaru, druhu a počtu FP 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 FP sa pokiaľ možno používa čo najjednoduchší, kvôli zníženiu výpočtovej náročnosti pri ich vyhodnocovaní; zvyčajne sú to tzv. trojuholníkové alebo lichobežníkové FP.

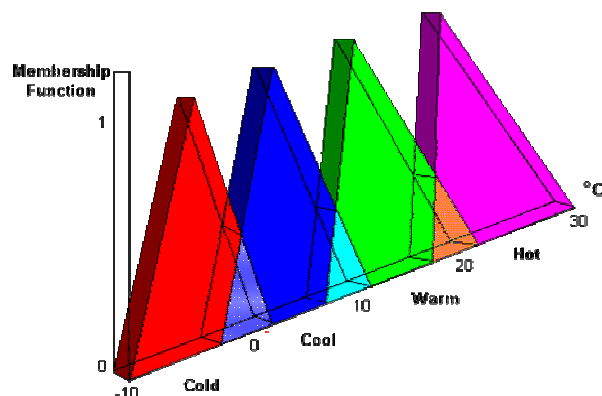
Ako príklad funkcie príslušnosti uvádzam neostré množiny studená, chladná, teplá, horúca. Každá z termov je definovaná funkciou príslušnosti na určitom intervale teplôt v stupňoch Celzia.



**Fig. 1 : Bivalent Sets to Characterize the Temp. of a room.**

Obr.1:Množiny charakterizujúce teploty v miestnosti

Po fuzzyfikácii dostaneme tento graf:



**Fig. 2 - Fuzzy Sets to characterize the Temp. of a room.**

Obr.2: Fuzzy množiny charakterizujúce teploty v miestnosti

Ako vidieť na obrázku 2, interval teplôt predstavuje univerzum, lingvistická premenná je teplota, lingvistické hodnoty sú studená, chladná, teplá, horúca a funkcia príslušnosti je trojuholníková.

## 2.3. Fuzzyfikácia

Proces priradovania meraných hodnôt vstupujúcich veličín do fuzzy množín pomocou funkcií príslušnosti sa označuje ako fuzzyfikácia. Hlavné aplikačné použitie fuzzy množín spočíva zrejme v tom, že pomocou nich môžeme interpretovať význam vágných jazykových výpovedí, tieto výpovede kvantifikovať a ich vzťahy formalizovať.



## 2.4. Defuzzyfikácia

Výsledkom činnosti bloku rozhodujúcich pravidiel je súbor funkcií príslušnosti pre jednotlivé termy lingvistických premenných. Pre praktické použitie fuzzy aproximácii je treba priradiť výstupným lingvistickým premenným ostrú hodnotu v prípustnom rozsahu. Tento proces „aproximácie neostrých termov“ ostrou hodnotu sa nazýva *defuzzyfikácia*.

## 2.5. Modelovanie procesov

Základnými pojmami pri identifikácii systémov sú reálny objekt a jeho model. Pod pojmom reálny objekt budeme mať na mysli originál, reálne zariadenie, predmet objektívnej reality, na ktorom je možné vykonať určité pozorovania za účelom poznania relácií v ňom prebiehajúcich.

Pod modelom reálneho objektu budeme rozumieť v zásade iný systém, ktorý nejakým spôsobom napodobňuje reálny originál. Pre naše účely budeme uvažovať matematický model systému, ktorý v tvare algebraických, diferenciálnych, diferenčných rovníc alebo pravdepodobnostných vzťahov kvantifikuje relácie medzi jednotlivými veličinami v reálnom objekte.

Klasifikovaním systémov podľa prijatých matematických modelov sa zaoberá modelovanie procesov.

Všeobecné aspekty výberu modelu:

- výber typu modelu - rozlišujeme medzi lineárnymi a nelineárnymi, stavovými a vstupno – výstupnými modelmi,
- výber veľkosti modelu – rád stavového modelu, stupne odhadovaných polynómov, atď...

## 2.6. Fuzzy modely

### *Model typu Mamdani*

Model typu Mamdani 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:

Ak  $x$  je  $A$  a  $y$  je  $B$  potom  $z$  je  $C$

Aj vstupy, aj výstup sú opísané pomocou fuzzy množín.

### *Model typu Sugeno*

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, alebo TSK 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:

Ak  $x$  je  $A$  a  $y$  je  $B$  potom  $z = f(x, y)$

Sugenovské modely možno veľmi jednoducho použiť najmä pre aproximáciu nelineárnych závislostí. Práca s nimi je pohodlná, najmä pokiaľ poznáme príslušné pravidlá, čo tiež predpokladá, že sme už spravili rozklad stavového priestoru na príslušný počet buniek charakterizovaných fuzzy zvolenými fuzzy množinami.

Fuzzy Toolbox Matlabu obsahuje štruktúru Anfis, ktorej výhodou je, že jej výsledkom je práve model typu Sugeno – neuro - fuzzy model. Práve preto som pracovala s týmto typom modelu a nie typom Mamdani.

### 3. PRAKTICKÁ ČASŤ

#### 3.1. Chemické reaktory

Chemické reaktory patria k základným zariadeniam chemických technológií. Z hľadiska matematického opisu a požiadaviek na riadenie sú tieto systémy veľmi zložité. Vyplýva to zo skutočnosti, že okrem fyzikálnych javov, ako je prúdenie, vedenie tepla, difúzia, prestup tepla prebiehajú v chemickom reaktore aj javy chemické, ktoré sú zvyčajne veľmi zložité. Preto sa matematické modely chemických reaktorov dajú získať len s použitím mnohých zjednodušujúcich predpokladov. Jedným z nich je linearizácia nelineárnych vzťahov, pričom linearizovaný model je použiteľný len v okolí pracovného režimu, kde sa linearizácia robila.

Podstata linearizácie nelineárnych rovníc spočíva v predpoklade, že veličiny procesu sa menia tak, že odchýlky od ustáleného stavu sú v čase dostatočne malé. Pri linearizácii sa používa rozvoj do Taylorovho modelu a potom sa uvažujú len lineárne členy radu.

Na vytvorenie môjho modelu stačila  $s$  – funkcia a nelineárne (diferenciálne rovnice). Uvažujme schému s  $s$  - funkciou:

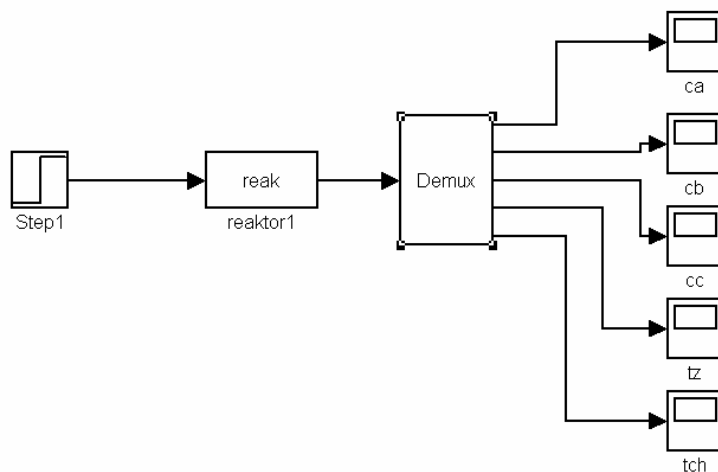
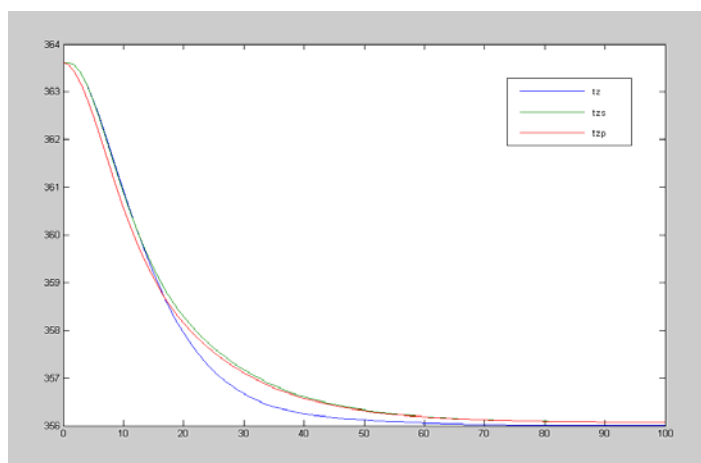


Schéma 1: Model chemického reaktora

Pomocou schémy 1 môžeme zisťovať ako sa menia koncentrácie, teplota reakčnej zmesi a teplota chladiacej zmesi ak na vstupe je skoková zmena prietoku chladiacej zmesi.

Na obr. 3 je priebeh teploty reakčnej zmesi pri spomenutej skokovej zmene.



Obr.3: Priebeh teploty reakčnej zmesi pri skokovej zmene prietoku

chladiacej zmesi.

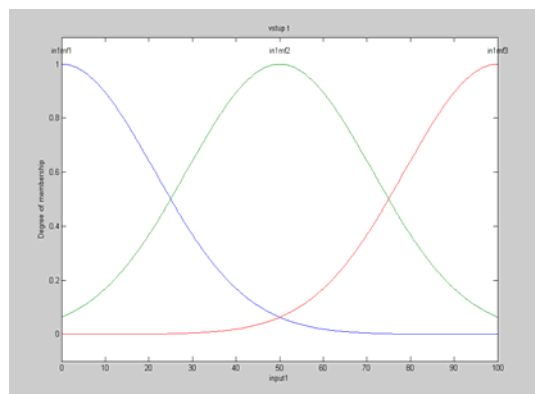
modrá čiara- model reaktora: s-funkcia, zelená čiara– stavový popis, červená čiara- prenos

### 3.2. Fuzzy model chemického reaktora

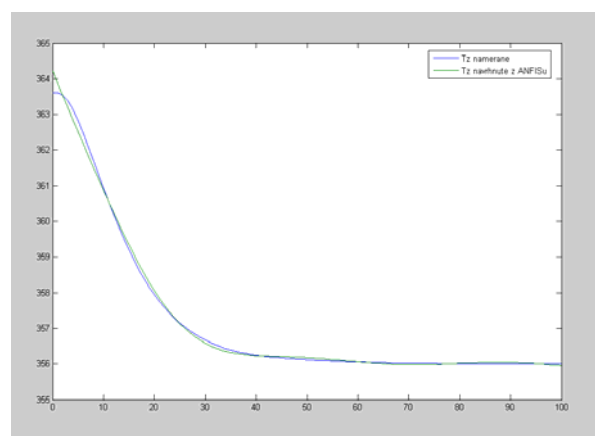
Má význam zväčšovať počet funkcií príslušnosti?

Má tvar funkcií príslušnosti vplyv na identifikáciu?

Na obr. 5 je neuro-fuzzy model typu Sugeno pri zadaní počtu funkcií príslušnosti (PFP) 3 a typu funkcie príslušnosti gaussmf, na obr. 4 je graf funkcie príslušnosti typu gaussmf a PFP rovný 3:



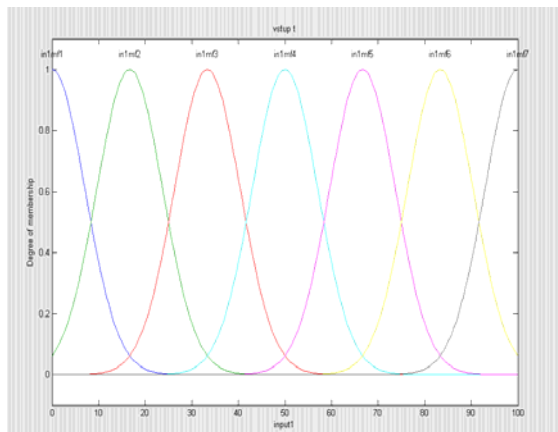
Obr.4: Graf funkcie príslušnosti typu gaussmf a PFP rovný 3



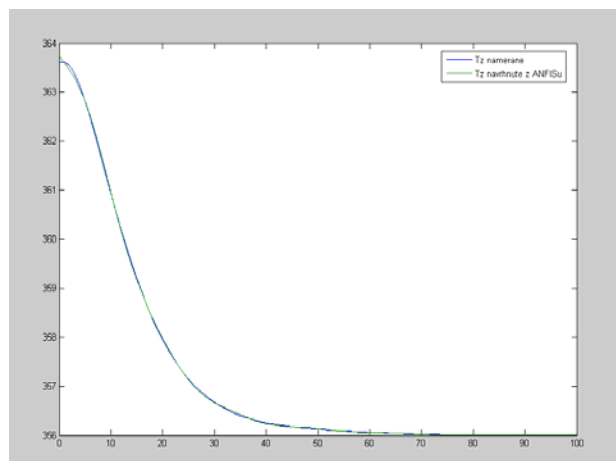
Obr. 5: Neuro – fuzzy model typu Sugeno

Ako vidieť na obrázku 5, modrá čiara predstavuje namerané hodnoty Tz (teplota zmesi) a zelená čiara predstavuje hodnoty Tz navrhnuté Anfis-om.

Zadaním rovnakého typu funkcie príslušnosti, ale zvýšením PFP na hodnotu 7 dostaneme nasledujúce grafy:



Obr.6:Graf funkcie príslušnosti typu gaussmf a PFP rovný 7



Obr. 7: Neuro – fuzzy model typu Sugeno

Pre tento uvažovaný priebeh nemá zvyšovanie PFP takmer žiadny význam. Pri vyššom PFP je jediný rozdiel v lepšom začiatočnom nábehu.

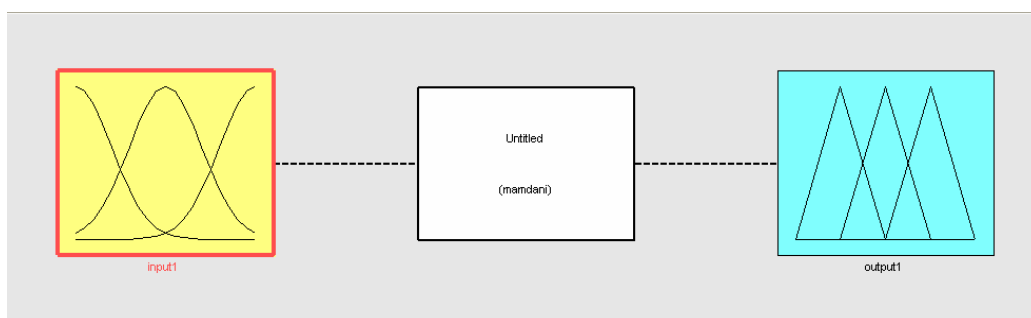
Pri porovnaní s jednoduchšou a bežnejšou funkciou príslušnosti trimf som zistila, že rozdiel je minimálny a nemá význam pre presnosť uvažovaného modelu.

### 3.3. Vytvorenie fuzzy modelu

Projektovanie modelov vyžaduje definovať vstupné a výstupné premenné, ich rozsahy, funkcie príslušnosti a ich parametre, zadávanie inferenčných a rozhodovacích pravidiel, nastavenie metód fuzzyfikácie a defuzzyfikácie.

#### 3.3.1. FIS Editor – Inferenčný systém fuzzy

Pre tieto požiadavky môžeme vyjadriť štruktúru modelu blokovo podľa obr. 8 zadaním príkazu >>fuzzy v programe Matlab.



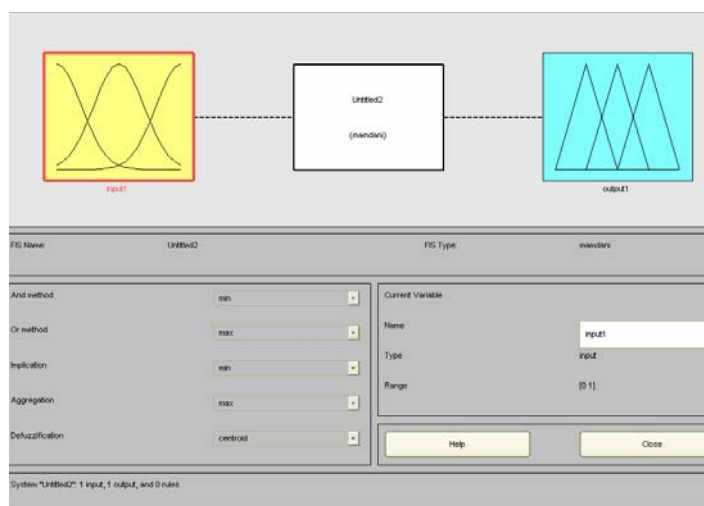
Obr. 8: FIS Editor

Túto štruktúru budeme označovať ako Fuzzy Interference System FIS – Inferenčný systém fuzzy.

FIS tvoria 3 editory: *FIS Editor* (editor inferenčného systému) – (obr.9 –10), *Membership Function Editor* (editor funkcií príslušnosti) – (obr.11-12), *Rule Editor* (editor pravidiel) – (obr.13) a dve zobrazovania – *Rule*-(obr.14–15) a *Surface Viewer* (obr.16) (grafické zobrazovanie procesu inferencie a plochy ohraničujúce priestor generovaných akčných zásahov).

Zo základnej ponuky programu Matlab si môžeme vybrať dva typy modelov:

### ***Model typu Mamdani***

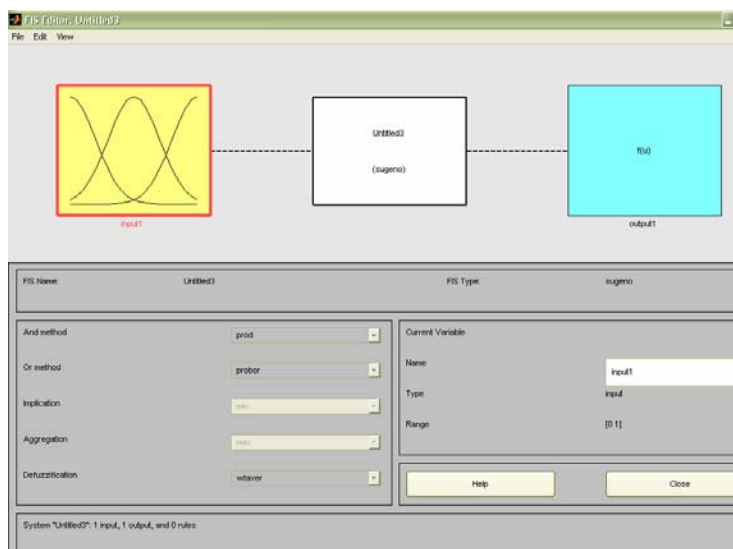


Obr. 9: FIS Editor pre model typu Mamdani

O modeli typu Mamdani som hovorila už v kapitole 2.7. Jeho charakteristickou vlastnosťou je fuzzy výstup.

### **Model typu Sugeno**

FIS štruktúra pre tento typ modelu je zobrazená na obrázku 10.



Obr. 10: FIS Editor pre model typu Sugeno

Pracovala som s modelom typu Sugeno, pretože ako som už spomínala v kapitole 2.4.1., tento model má na výstupe ostré (crisp) hodnoty. Je výstupom z Anfis štruktúry.

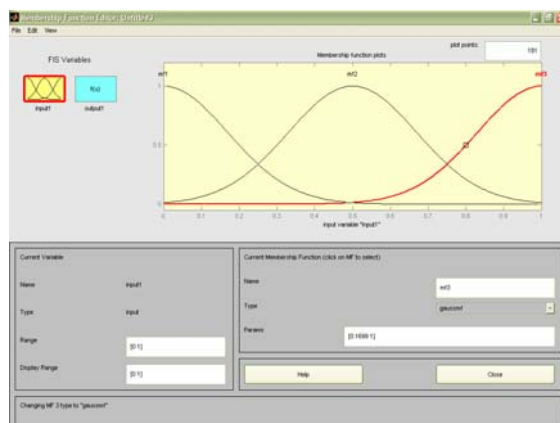
Anfis pre „tréning“ FIS členov na určenie parametrov funkcií príslušnosti používa kombináciu metódy najmenších štvorcov a metódy spätného šírenia. Prostredníctvom týchto dvoch metód je Anfis schopný natrénovať údaje, čo v prvom rade zjednodušuje prácu pri tvorbe jednotlivých modelov.

Hlavné menu FIS Editoru na obr. 10 obsahuje roletové menu *File*, *Edit*, *View*, ktoré umožňujú ukladanie a volanie súborov a editáciu fuzzy systému. V ponuke Edit je možné pridaním alebo odobratím určiť počet vstupov, maximálne 4 vstupy a 1 výstup.

V grafickom okne na obr. 10 sú v princípe zobrazované tromi ikonami – blokmi: vstupné premenné, typ inferencie FIS, výstupné premenné.

### 3.3.2. Membership Funktion Editor – Editor funkcií príslušnosti

Dvojitým kliknutím na vybranú vstupnú premennú reprezentovanú obrázkom je možné prejsť do Membership Funktion Editor – obr. 11. Spustí sa tiež cez roletové okno Membership Funktion Editor.



Obr. 11: Editor funkcií príslušnosti pre vstup



Obr. 12: Editor funkcií príslušnosti pre výstup

V okne vľavo dole sú uvedené: meno, typ, rozsah a rozsah displeja označených heslami v anglickom jazyku vo vymenovanom poradí. V tomto okne je možné zadávať potrebné rozsahy. Na pravej strane možno voliť funkcie príslušnosti pre vstupnú alebo výstupnú premennú. V mojom prípade je to opäť typ gaussmf.

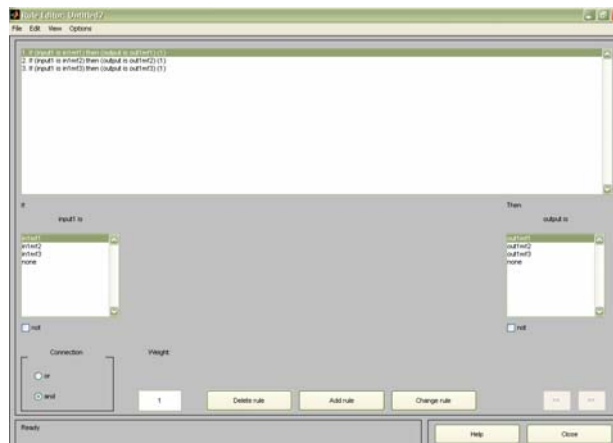
Po dvojitom kliknutí na výstupnú premennú dostanem opäť okno v ktorom môžem zadávať parametre pre výstup (obr.12). Rozdiel je v type výstupu. V Matlab-e môžeme vybrať typ lineárny alebo konštantu. Ja som pracovala s typom lineárnym, ale na obrázku 21 vidieť priebeh s typom konštanty.

### 3.3.3. Rule Editor – Editor pravidiel

Spustí sa vo FIS Editore pomocou roletového okna View – Edit Rules (obr. 12). Obsahuje editačné a zobrazovacie pole. V tomto poli je možné pravidlá priamo editovať ručne alebo použiť zadané tlačidlá:

- Delete rule – vymaže pravidlo,
- add rule - pridá pravidlo,
- change rule – zmení pravidlo.



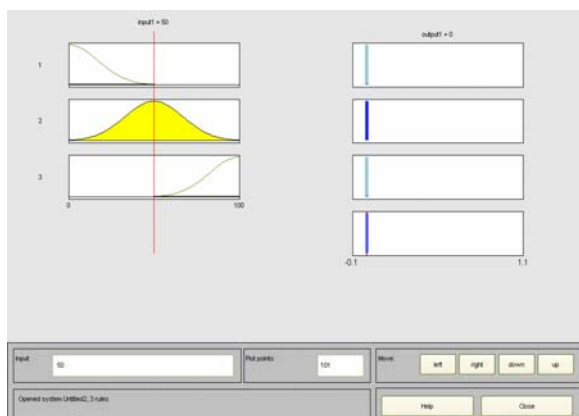


Obr. 13: Editor pravidiel

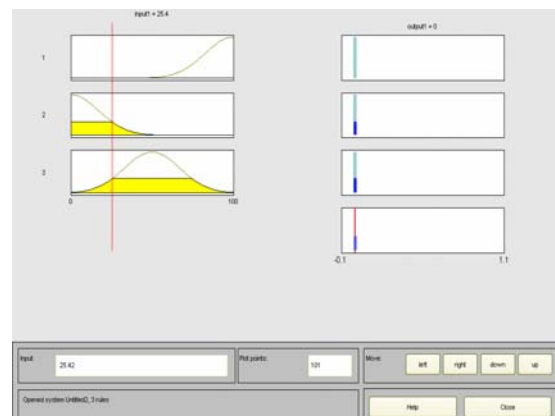
### 3.3.4. Rule Viewer – Grafické zobrazenie procesu inferencie

Aktivuje sa pomocou roletových menu View – Rule Viewer. Obsahuje všetky pravidlá, tvary funkcií príslušnosti vstupov, výstupov a ich inferencie.

Pravidlá prislúchajúce počtu funkcií príslušnosti 3 pre 2 rôzne hodnoty výstupov vyzerajú takto:



Obr. 14: Grafické zobrazenie procesu inferencie

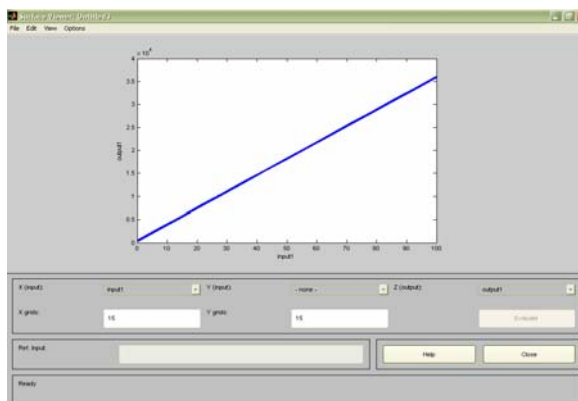


Obr. 15: Grafické zobrazenie po prepočítaní pravidiel

Červená čiara v ľavej časti charakterizujúcej vstup nám zobrazuje výslednú ostrú hodnotu. Jej pohybom zistíme, ktoré funkcie príslušnosti sa prekrývajú a ako sa potom zmení výstup – obr. 15.

### 3.3.5. Surface Viewer – Plochy ohraničujúce priestor generovaných akčných zásahov

Aktivuje sa pomocou roletových menu View – Surface Viewer. Zobrazuje priestor výstupnej veličiny v závislosti od vstupných veličín.



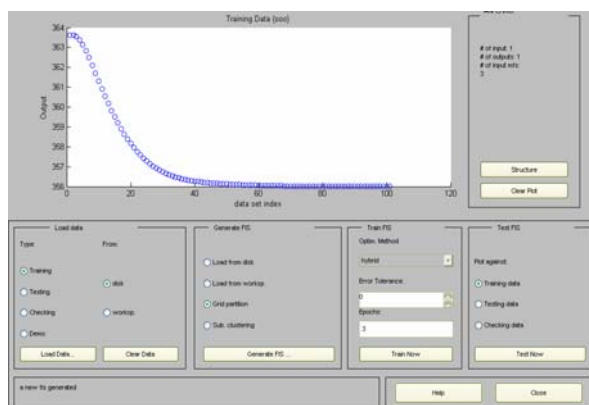
Obr. 16:  $y = f(u)$

## 3.4. Anfis

Anfis využíva schopnosť hybridného určenia algoritmov k identifikácii parametrov funkcií príslušnosti jednoduchého výstupu, ako je Sugenov typ fuzzy systémov.

Pre „tréning“ FIS členov na určenie parametrov funkcií príslušnosti sa používa kombinácia metódy najmenších štvorcov a metódy spätného šírenia.

Zadaním príkazu `>>anfisedit` v Matlab-e sa otvorí okno na obr. 17.

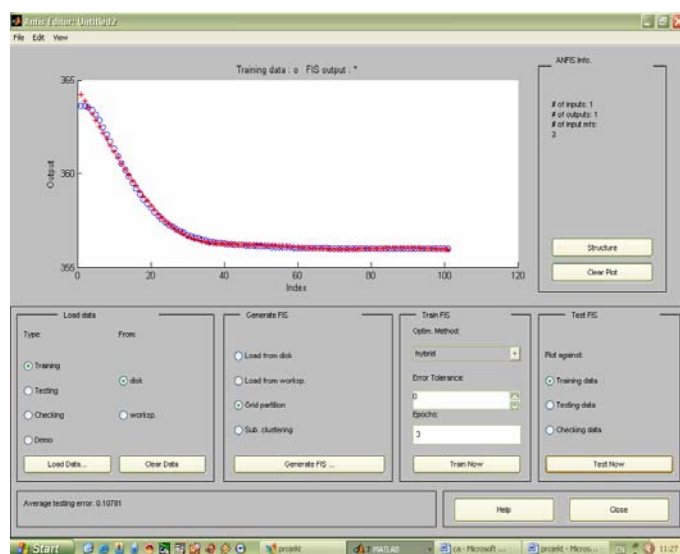


Obr. 17: Anfis editor

Stlačením Load data možno načítať uložené údaje – v mojom prípade hodnoty Tz.

Cez generate FIS si presne zvolíme počet funkcií príslušnosti rovný 3, typ funkcie príslušnosti vo vstupe gaussmf a typ funkcie príslušnosti vo výstupe, ktorý bol v mojom prípade lineárny. Po natrénovaní a testovaní údajov je Anfis schopný vydať testovacie údaje podľa FIS vzoru.

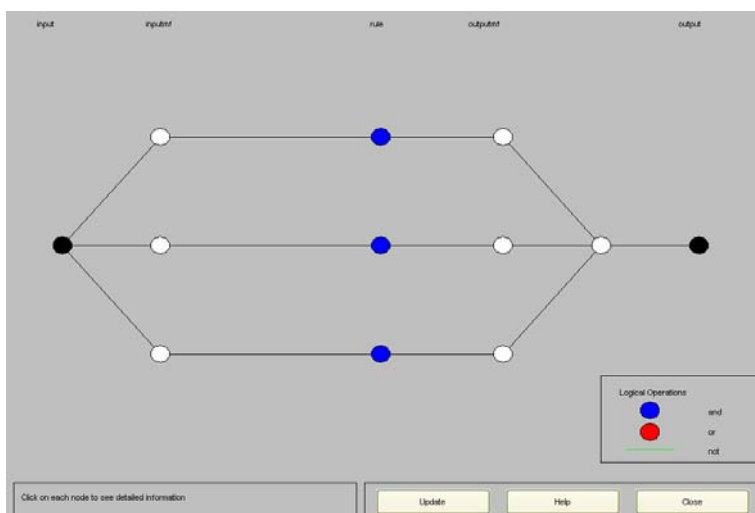
Táto zhoda je zobrazaná červenou výplňou na obrázku 18.



Obr. 18: Trénovacie údaje oproti FIS vzoru

Výsledná Anfis štruktúra ( adaptívna sieť ) je nasledovná:

vstup    vrstva1                      vrstva2    vrstva3    vrstva4    vrstva5



Obr.19: Výsledná Anfis štruktúra,  $y = f(u)$

Vrstva 1 – Každý uzol  $i$  tejto vrstvy je adaptívny. Výstupom je hodnota príslušnosti vstupu  $x$  k úrovni  $A_i$  lingvistickej premennej.

Vrstva 2 – Každý uzol tejto vrstvy je neadaptívny. Násobí vstupné signály a vysieľa ich súčin. Reprezentuje silu pravidla.

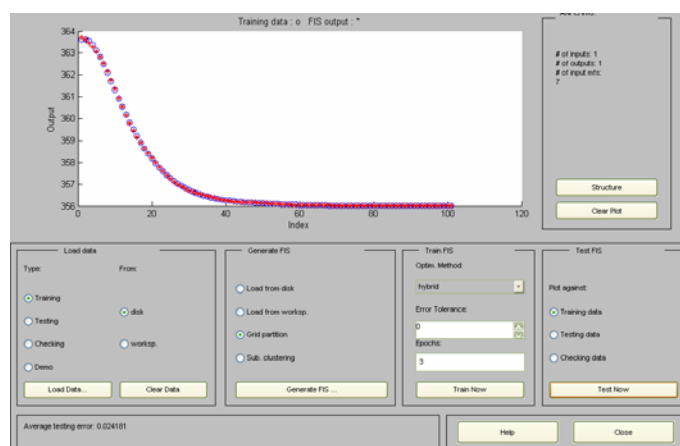
Vrstva 3 – Každý uzol tejto vrstvy je neadaptívny a počíta pomer sily i-teho pravidla k sume všetkých síl pravidiel.

Vrstva 4 - Uzol tejto vrstvy je adaptívny.

Vrstva 5 – Neadaptívny uzol plniaci funkciu sčítania vstupných signálov. Ak označíme  $In_n$  počet vstupov a  $Mf_i$  počet funkcií príslušnosti i-teho vstupu, potom pre počet všetkých možných pravidiel  $Rule_n$ , možno písať

$$Rule_n = \prod_{i=1}^{In_n} Mf_i$$

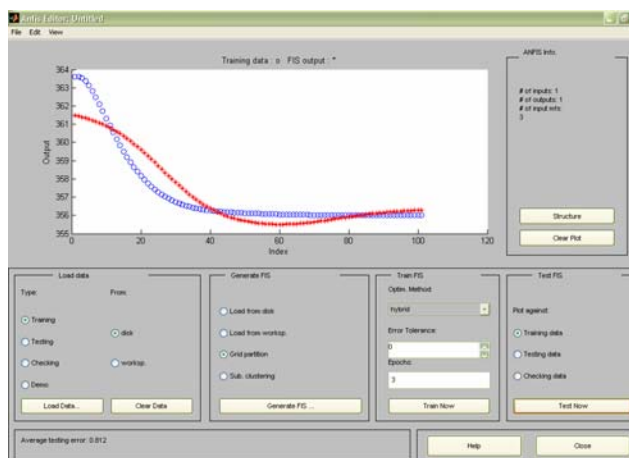
Pre PFP rovný 7 budú natréňované údaje takéto:



Obr.20: Trénovanie údajov oproti FIS vzoru

V uvažovanom prípade pri porovnaní jednotlivých grafov vidíme iba nepatrný rozdiel v nábehu pri PFP 7. Inak zvýšenie PFP nemá vplyv ani na pravidlá ani štruktúru modelu.

Ak vo výstupe zadám ako typ funkcie príslušnosti konštantu, výsledok z Anfis-u bude vyzerat' nasledovne:



Obr.21: Trénovacie údaje oproti FIS vzoru – typ funkcie vo výstupe je konštanta

## 4. TVORBA FUZZY MODELOV

Na ilustráciu postupu pri tvorbe fuzzy modelov s využitím Fuzzy Toolboxu sú uvedené 3 príklady.

### 4.1. Prvý príklad

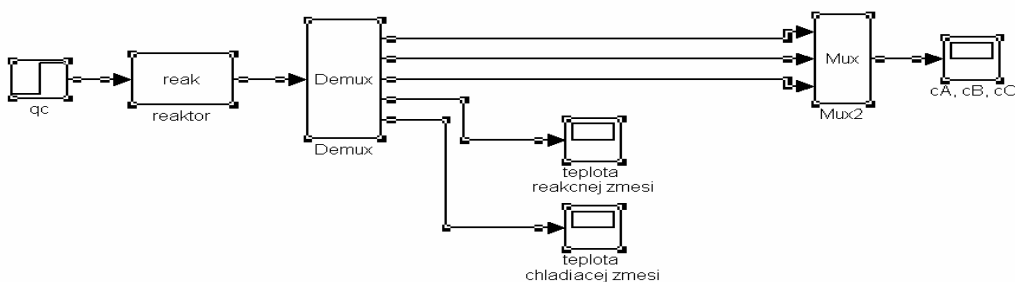
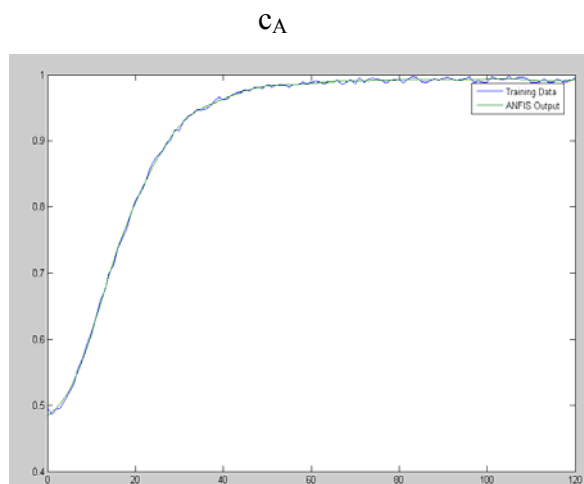


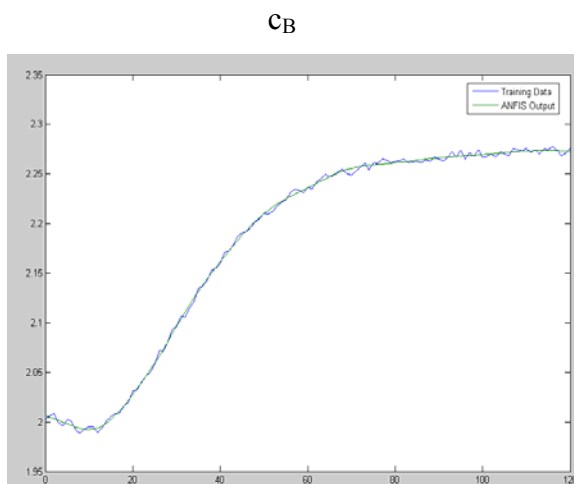
Schéma 2: Schéma chemického reaktora

V schéme 2 získame priebehy  $c_A$ ,  $c_B$ ,  $c_C$ ,  $t_Z$ ,  $t_{CZ}$  ako odozvy na skokovú zmenu  $q_c$ . Výsledky môžu byť použité ako tréningové údaje pre návrh modelu typu Sugeno. Pre tento účel je vytvorený krátky program *anf.m* uvedený v prílohe. Pri návrhu môžeme voliť počet a tvar funkcií príslušnosti.

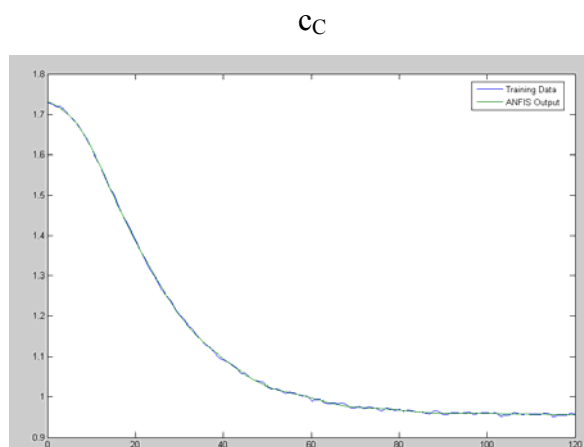
Pre 7 funkcií príslušnosti a typ *gaussmf* sú výsledky uvedené na obr. 22 - 26.



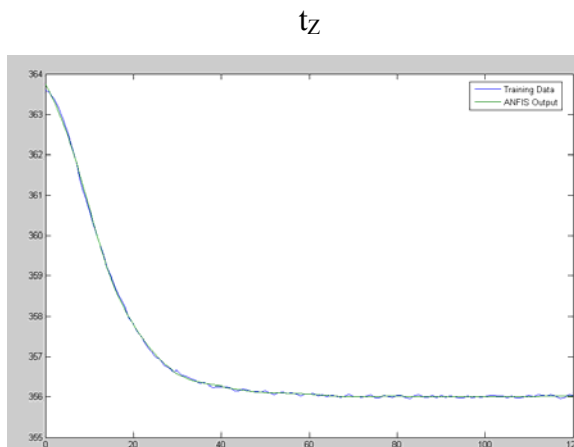
Obr.22: Priebeh koncentrácie  $c_A$



Obr. 23: Priebeh koncentrácie  $c_B$

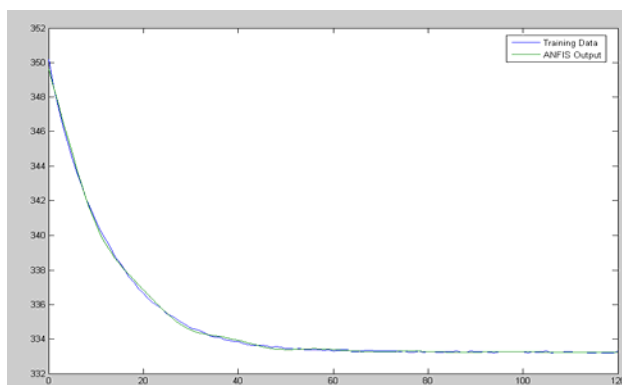


Obr. 24: Priebeh koncentrácie  $c_C$



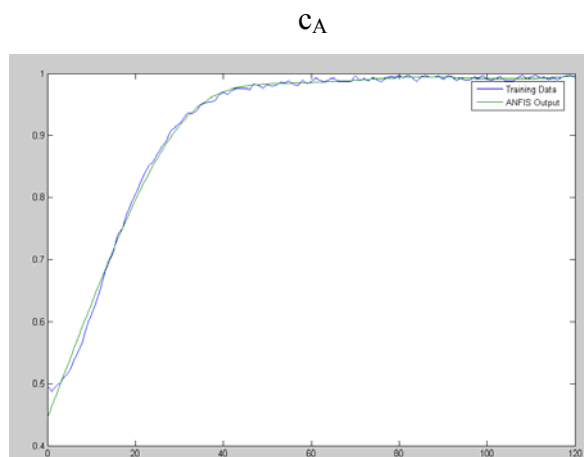
Obr. 25: Priebeh teploty reakčnej zmesi

$t_{CZ}$

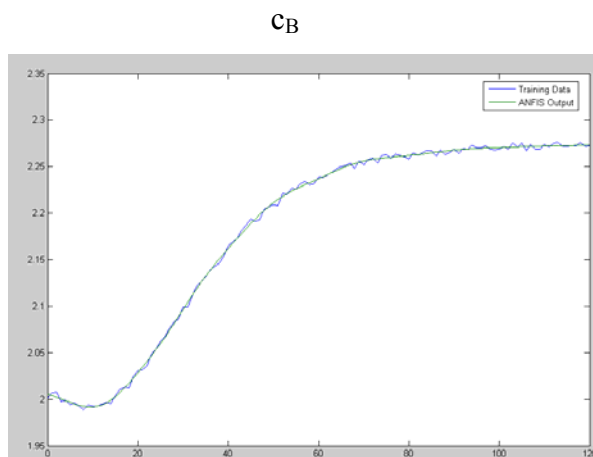


Obr. 26: Priebeh teploty chladiacej zmesi

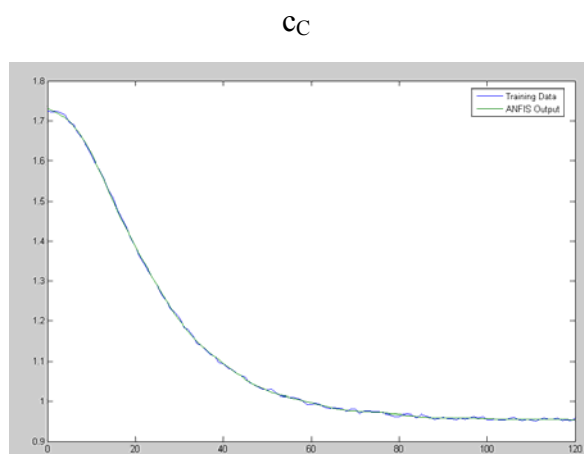
Pre počet funkcií príslušnosti rovný 3 sú výstupy pre priebehy  $c_A$ ,  $c_B$ ,  $c_C$ ,  $t_Z$ ,  $t_{CZ}$  ako odozvy na skokovú zmenu  $q_C$  uvedené na obr. 27 - 31.



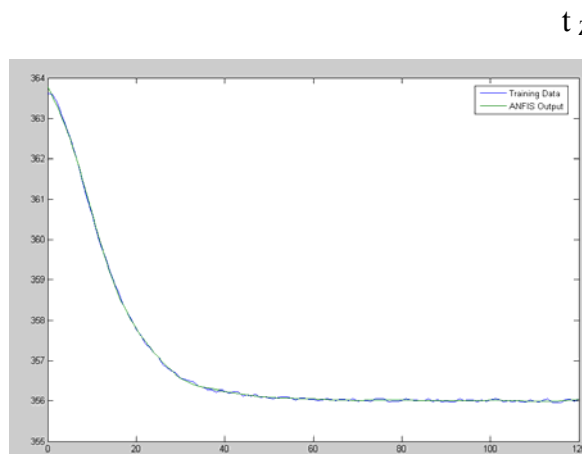
Obr. 27: Priebeh koncentrácie  $c_A$



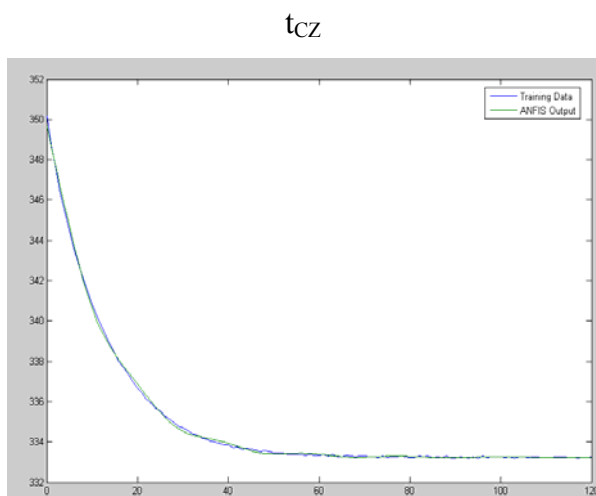
Obr.28: Priebeh koncentrácie  $c_B$



Obr. 29: Priebeh koncentrácie  $c_C$



Obr. 30: Priebeh teploty reakčnej zmesi

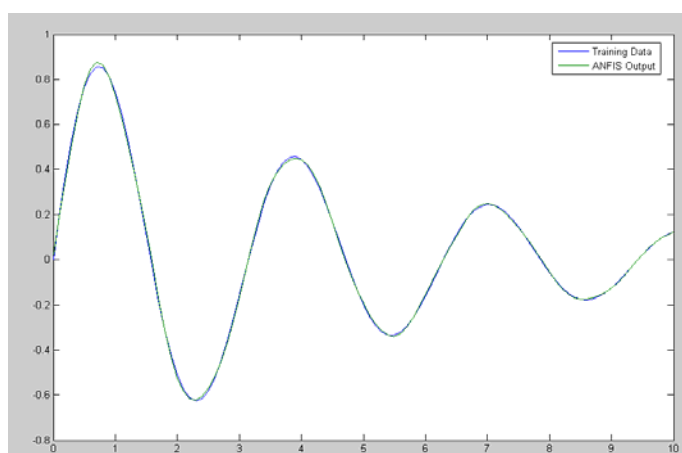


Obr. 31: Priebeh teploty chladiacej zmesi

## 4.2. Druhý príklad

V tomto prípade zisťujeme ako vplýva veľkosť počtu funkcií príslušnosti na zhodu modelu. Pre tento účel je vytvorený krátky program *anfprklad.m* uvedený v prílohe. Pri návrhu môžeme opäť voliť typ funkcie príslušnosti a PFP.

Pre PFP rovný 9, typ funkcie príslušnosti *gaussmf* a funkciu  $y = \sin(2*x)/\exp(x/5)$  je výstup zobrazený na obrázku 32.



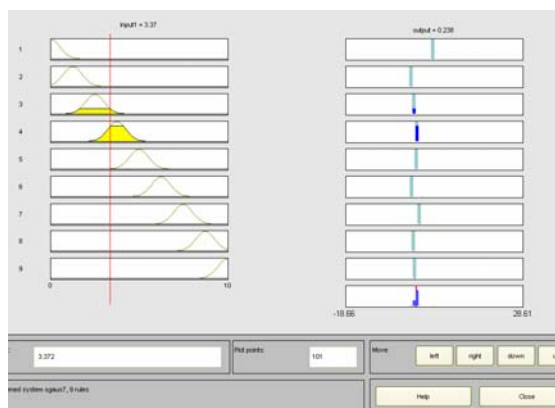
Obr. 32: Výstup z Anfis štruktúry

Modrá krivka predstavuje údaje na tréning, zelená predstavuje výstup z anfis, čiže natréňované údaje. Model stíha sledovať priebeh.

Pravidlá prislúchajúce počtu funkcií príslušnosti 9 pre 2 rôzne hodnoty výstupov vyzerajú takto:



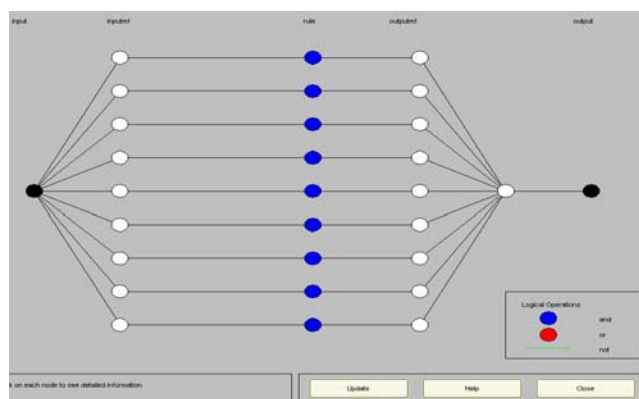
Obr. 33: Grafické zobrazenie procesu inferencie



Obr. 34: Grafické zobrazenie po prepočítaní pravidiel

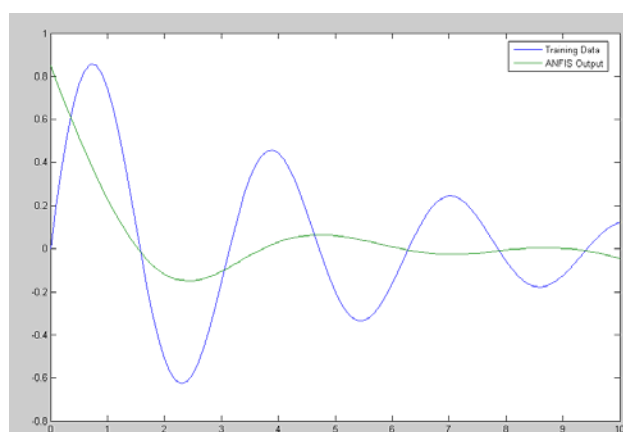


Výsledná štruktúra Anfis pre 1 vstup a 1 výstup pre uvedený príklad:



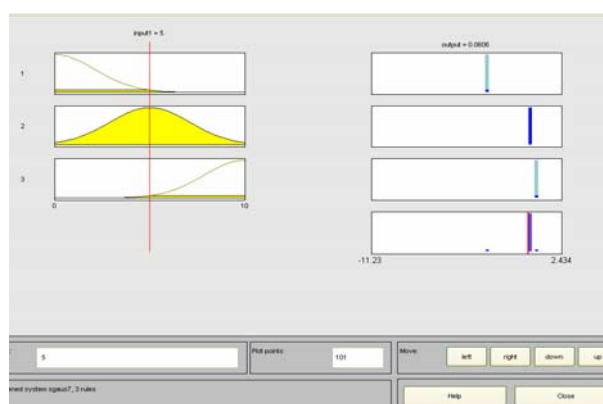
Obr. 35:  $y = f(u)$

Ak počet funkcií príslušnosti zmeníme na hodnotu 3, grafy zmenia svoje priebehy:

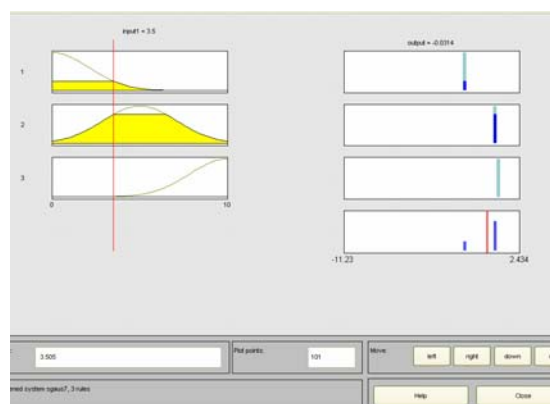


Obr. 36: Výstup z Anfis štruktúry

Pravidlá:

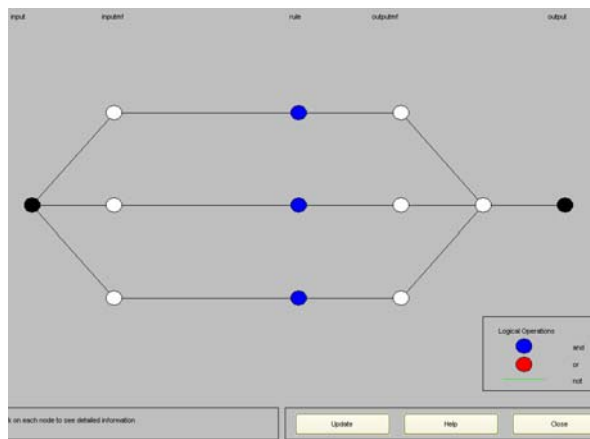


Obr. 37: Grafické zobrazenie procesu inferencie



Obr. 38: Grafické zobrazenie po prepočítaní pravidiel

Výsledná štruktúra:



Obr. 39:  $y = f(u)$

Základná otázka bola, ako vlastne vplýva zmena počtu funkcií príslušnosti na priebeh a výslednú štruktúru modelu?

V uvažovaných prípadoch vyššia hodnota PFP znamená väčší počet pravidiel, a tým je podmienená lepšia presnosť modelu, viz obr. 32. Celý priebeh takmer identicky kopíruje trénované údaje.

Naopak, pri nižšej hodnote PFP je priebeh horší, zhoda s trénovanými údajmi je takmer minimálna, viz obr. 36.

Štruktúra modelu sa ani v jednom prípade nemení, je rovnaká. Štruktúra sa môže meniť iba v prípade viacerých vstupov, resp. výstupov. V našom prípade sme mali jeden vstup a jeden výstup, takže zmena nenastala žiadna.

Schválne som zvolila zložitejšiu funkciu, na ktorej je vidieť, že pri vyššom PFP je presnosť modelu veľmi dobrá.

### 4.3. Tretí príklad

V tomto prípade uvažujeme 2 vstupy a 1 výstup.

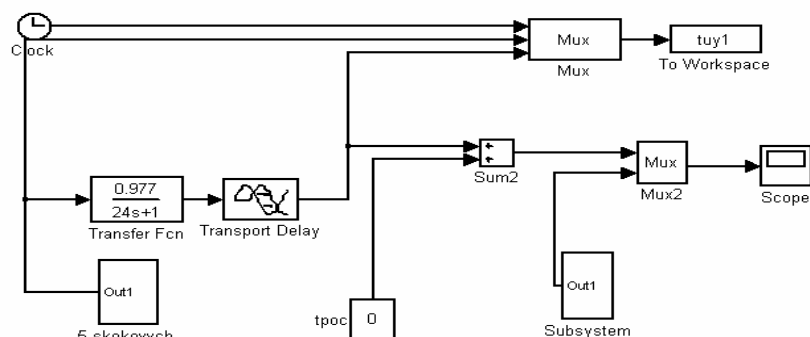
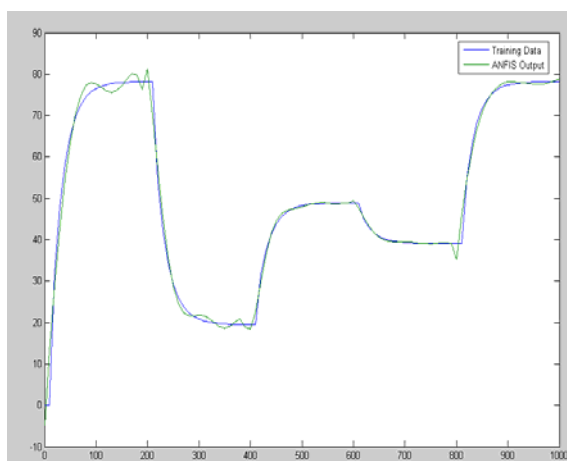


Schéma 3: Schéma systému

Pre tento účel je vytvorený krátky program *anfu.m* uvedený v prílohe.

Základom práce s tretím modelom, ktorý je opísaný ako sústava 1. rádu s dopravným oneskorením je schéma 3, kde neuvažujeme iba časový priebeh výstupných veličín, ale na vstupe je 5 skokových zmien vstupnej veličiny  $u$ . Výsledný priebeh  $y$  je zobrazený na obrázku 40.



Obr. 40: Výsledný priebeh  $y$

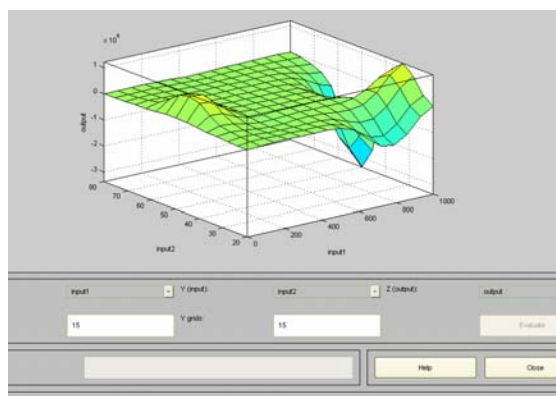
Keďže mám dva vstupy a pre každý vstup uvažujeme 5 funkcií príslušnosti typu *gaussmf*, počet pravidiel bude  $5 \times 5 = 25$  pravidiel. Jednotlivé pravidlá sú priložené v prílohe.

Zobrazenie pravidiel:



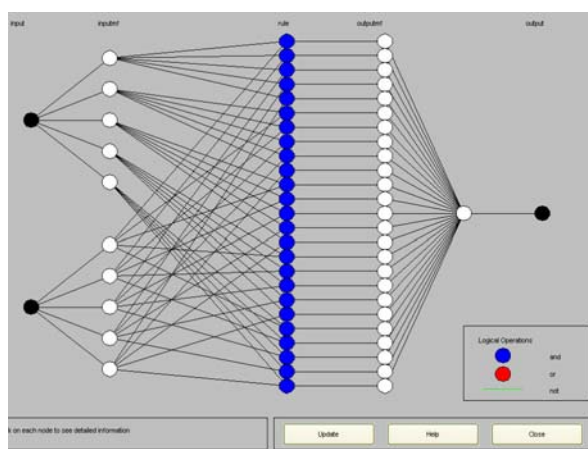
Obr. 41: Grafické zobrazenie inferencie

Výsledný povrch je zobrazený trojrozmerné.



Obr.42:  $y = f(u_1, u_2)$

Štruktúra pre dva vstupy a jeden výstup je nasledovná:



Obr. 43: Výsledná štruktúra pre  $y = f(u_1, u_2)$

Jednotlivé prepojenia medzi vstupmi a pravidlami sú omnoho zložitejšie ako pri 1 vstupe.

Uvedené modely ilustrujú vplyv počtu funkcií príslušnosti na presnosť modelu, ukazujú ako sa mení model pri viacerých vstupoch.

## 5. ZÁVER

Názov môjho bakalárskeho projektu znie: Fuzzy modelovanie dynamiky chemického procesu v prostredí Matlab. To znamená, že prácu som robila prostredníctvom tohto programu.

Práca mi bola uľahčená tým, že Anfis je priamo súčasťou Matlab-u. Vaužitím štruktúry Anfis som priamo získala neuro fuzzy model.

V mojej práci som zisťovala ako vplýva počet funkcií príslušnosti na priebeh a štruktúru uvažovaných modelov.

Tvar FP sa pokiaľ možno používa čo najjednoduchší, kvôli zníženiu výpočtovej náročnosti pri ich vyhodnocovaní; zvyčajne sú to tzv. trojuholníkové (trimf) alebo lichobežníkové (trapmf) FP. Sú jednoduché a zvyčajne postačujú. Pri zložitejších modeloch sa používa napr. gaussmf. Tento typ som používala aj ja.

Počet funkcií príslušnosti tiež závisí od zložitosti priebehov. Ak máme zložitejší priebeh, treba PFP zvýšiť.

Pri jednoduchších priebehoch to je relatívne jedno. V tomto prípade ide iba o zlepšenie nábehu pri vyššom PFP, ako je tiež vidieť pri porovnaní obr. 5 a 7.

Aj zložitosť funkcií býva rozhodujúcim faktorom pri výbere PFP. Napr. podľa mojich skúseností pri použití funkcie  $y = \sin(2 \cdot x) / \exp(x/5)$  je výhodnejšie použiť vyšší PFP ako je vidieť pri porovnaní obr. 32, kde je PFP rovný 9 s obrázkom 36, kde je PFP rovný 3. Z toho vyplýva, čím vyšší PFP, tým je viac pravidiel a lepšia presnosť modelu.

Ak máme model s dvoma vstupmi a jedným výstupom, počet pravidiel sa zdvojnásobí. Zväčša to býva veľmi komplikované a pre ľudský rozum ťažké na spracovanie. Výrazne sa mení výsledná štruktúra modelu, ktorá je zložitejšia kvôli prepájaniu jednotlivých vstupov s pravidlami.

## 6. LITERATÚRA

- [1] M. Fikar a kol. Identifikácia systémov. Vydavateľstvo STU v Bratislave, 1999.
- [2] J. Mikleš a kol. Riadenie technologických procesov – Modelovanie procesov a základy riadenia. Vydavateľstvo STU v Bratislave, 1994.
- [3] O. Modrlák. Fuzzy řízení a regulace.2002
- [4] P. Vysoký. Fuzzy řízení. Vydavateľstvo ČVUT, Praha 1996
- [5] <http://www.dcssoft.sk/asc/celjuska/fuzzy/index.phtml>

## 7. PRÍLOHY

### *Príloha č.1 – pravidlá pre 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=[0 100]
NumMFs=7
MF1='in1mf1': 'gaussmf',[7.08 -0.003]
MF2='in1mf2': 'gaussmf',[7.08 16.6]
MF3='in1mf3': 'gaussmf',[7.08 33.3]
MF4='in1mf4': 'gaussmf',[7.07 49.9]
MF5='in1mf5': 'gaussmf',[7.07 66.6]
MF6='in1mf6': 'gaussmf',[7.07 83.3]
MF7='in1mf7': 'gaussmf',[7.07 99.9]

[Output1]
Name='output'
Range=[356.00499 363.61]
NumMFs=7
MF1='out1mf1': 'linear',[-0.04 363.8]
MF2='out1mf2': 'linear',[-0.1 361.005]
MF3='out1mf3': 'linear',[-0.05 358.2]
MF4='out1mf4': 'linear',[-0.01 357.05]
MF5='out1mf5': 'linear',[-0.006 356.5]
MF6='out1mf6': 'linear',[-0.001 356.1]
MF7='out1mf7': 'linear',[0.0008 355.9]

[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
```



### **Príloha č. 2 – pravidlá pre prvý model**

po zadaní príkazu „type sgaus7cb“ do programu Matlab

```
[Input1]
Name='input1'
Range=[0 120]
NumMFs=7
MF1='in1mf1':gaussmf,[8.4 -0.001]
MF2='in1mf2':gaussmf,[8.4 20.004]
MF3='in1mf3':gaussmf,[8.5 40.0009]
MF4='in1mf4':gaussmf,[8.5 59.9]
MF5='in1mf5':gaussmf,[8.5 79.9]
MF6='in1mf6':gaussmf,[8.4 99.9]
MF7='in1mf7':gaussmf,[8.4 120.0]
```

```
[Output1]
Name='output'
Range=[1.9 2.2]
NumMFs=7
MF1='out1mf1':linear,[-0.0001 2.01]
MF2='out1mf2':linear,[0.005 1.9]
MF3='out1mf3':linear,[0.004 1.9]
MF4='out1mf4':linear,[0.002 2.09]
MF5='out1mf5':linear,[0.0006 2.2]
MF6='out1mf6':linear,[0.0002 2.2]
MF7='out1mf7':linear,[-0.0001 2.2]
```

```
[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
```

### **Príloha č. 3 – pravidlá pre tretí model**

```
[Input1]
Name='input1'
Range=[0 1000]
NumMFs=5
MF1='in1mf1':gaussmf,[106.1 -2.2]
MF2='in1mf2':gaussmf,[106.1 249.9]
MF3='in1mf3':gaussmf,[106.1 499.9]
MF4='in1mf4':gaussmf,[106.1 749.9]
MF5='in1mf5':gaussmf,[106.1 999.9]
```

```
[Input2]
Name='input2'
Range=[20 80]
```

```

NumMFs=5
MF1='in2mf1':gaussmf',[6.3 19.9]
MF2='in2mf2':gaussmf',[6.3 34.9]
MF3='in2mf3':gaussmf',[6.3 49.9]
MF4='in2mf4':gaussmf',[6.4 65.0]
MF5='in2mf5':gaussmf',[6.3 80.0]

```

[Output1]

Name='output'

Range=[0 78.1]

NumMFs=25

```

MF1='out1mf1':linear',[50.1 -318.7 -15.9]
MF2='out1mf2':linear',[98.9 -2.1 -0.6]
MF3='out1mf3':linear',[1552.6 285.3 5.7]
MF4='out1mf4':linear',[99.3 18.2 0.3]
MF5='out1mf5':linear',[-3.3 -0.4 -0.005]
MF6='out1mf6':linear',[12.2 -129.9 -6.4]
MF7='out1mf7':linear',[-37.8 -78.0 -2.02]
MF8='out1mf8':linear',[57.1 -425.2 -8.5]
MF9='out1mf9':linear',[5.1 -26.8 -0.5]
MF10='out1mf10':linear',[5.8 -10.7 -0.1]
MF11='out1mf11':linear',[21.6 -438.7 -21.9]
MF12='out1mf12':linear',[-9.4 125.1 2.4]
MF13='out1mf13':linear',[9.8 -101.6 -1.7]
MF14='out1mf14':linear',[2.2 -24.8 -0.3]
MF15='out1mf15':linear',[20.8 -219.7 -2.7]
MF16='out1mf16':linear',[-182.2 -68.8 -3.5]
MF17='out1mf17':linear',[-13.9 212.7 5.3]
MF18='out1mf18':linear',[34.0 -431.1 -8.1]
MF19='out1mf19':linear',[2.3 -34.1 -0.6]
MF20='out1mf20':linear',[0.4 -3.3 -0.04]
MF21='out1mf21':linear',[2.7 2.38 0.05]
MF22='out1mf22':linear',[154.2 235.1 5.82]
MF23='out1mf23':linear',[-414.4 61.7 1.7]
MF24='out1mf24':linear',[-31.0 -2.0 -0.0]
MF25='out1mf25':linear',[2.1 -1.004 -0.01]

```

[Rules]

```

1 1, 1 (1) : 1
1 2, 2 (1) : 1
1 3, 3 (1) : 1
1 4, 4 (1) : 1
1 5, 5 (1) : 1
2 1, 6 (1) : 1
2 2, 7 (1) : 1
2 3, 8 (1) : 1
2 4, 9 (1) : 1
2 5, 10 (1) : 1
3 1, 11 (1) : 1
3 2, 12 (1) : 1
3 3, 13 (1) : 1

```

```

3 4, 14 (1) : 1
3 5, 15 (1) : 1
4 1, 16 (1) : 1
4 2, 17 (1) : 1
4 3, 18 (1) : 1
4 4, 19 (1) : 1
4 5, 20 (1) : 1
5 1, 21 (1) : 1
5 2, 22 (1) : 1
5 3, 23 (1) : 1
5 4, 24 (1) : 1
5 5, 25 (1) : 1

```

#### **Príloha č. 4 – program anf.m pre prvý model**

```

c=[t ca cb cc], tz=[t tz], tch=[t tch] zo schemy reaksim
x=c(:,1);ca=c(:,2);cb=c(:,3);cc=c(:,4);tz=tz(:,2);tch=tch(:,2);
y=ca;trnData=[x y];
    numMFs = 3;
    mfType = 'gaussmf';
    epoch_n = 3;
    in_fisimat = genfis1(trnData,numMFs,mfType)
    out_fisimat = anfis(trnData,in_fisimat,3);
    writefis(out_fisimat,'sgaus7ca');
    %type sgaus7ca.fis
    ew=evalfis(x,out_fisimat);
    %save ew1.dat ew -ascii
    plot(x,y,x,ew);
    legend('Training Data','ANFIS Output'),pause
    %e=ew-y;plot(x,e);legend('chyba');pause
    %s1=sum(e)
    %s2=sum(e.*e)
    %s3=sum(abs(e))
    %ruleview('swgaus7a'),pause
    %surfview('swgaus7a')
    fisimat1=readfis('sgaus7ca.fis');

y=cb;trnData=[x y];
    numMFs = 7;
    mfType = 'gaussmf';
    epoch_n = 3;
    in_fisimat = genfis1(trnData,numMFs,mfType)
    out_fisimat = anfis(trnData,in_fisimat,3);
    writefis(out_fisimat,'sgaus7cb');
    %type sgaus7cb.fis
    ew=evalfis(x,out_fisimat);
    %save ew1.dat ew -ascii
    plot(x,y,x,ew);
    legend('Training Data','ANFIS Output'),pause
    fisimat2=readfis('sgaus7cb.fis');

y=cc;trnData=[x y];
    numMFs = 7;
    mfType = 'gaussmf';
    epoch_n = 3;
    in_fisimat = genfis1(trnData,numMFs,mfType)
    out_fisimat = anfis(trnData,in_fisimat,3);

```

```

writefis(out_fisimat,'sgaus7cc');
%type sgaus7cc.fis
ew=evalfis(x,out_fisimat);
%save ew1.dat ew -ascii
plot(x,y,x,ew);
legend('Training Data','ANFIS Output'),pause
fisimat3=readfis('sgaus7cc.fis');

```

```

y=tz;trnData=[x y];
numMFs = 7;
mfType = 'gaussmf';
epoch_n = 3;
in_fisimat = genfis1(trnData,numMFs,mfType)
out_fisimat = anfis(trnData,in_fisimat,3);
writefis(out_fisimat,'sgaus7tz');
%type sgaus7tz.fis
ew=evalfis(x,out_fisimat);
%save ew1.dat ew -ascii
plot(x,y,x,ew);
legend('Training Data','ANFIS Output'),pause
fisimat4=readfis('sgaus7tz.fis');

```

```

y=tch;trnData=[x y];
numMFs = 7;
mfType = 'gaussmf';
epoch_n = 3;
in_fisimat = genfis1(trnData,numMFs,mfType)
out_fisimat = anfis(trnData,in_fisimat,3);
writefis(out_fisimat,'sgaus7tch');
%type sgaus7tch.fis
ew=evalfis(x,out_fisimat);
%save ew1.dat ew -ascii
plot(x,y,x,ew);
legend('Training Data','ANFIS Output')
fisimat5=readfis('sgaus7tch.fis');

```

### **Príloha č.5 – program anfpríklad.m pre druhý model**

```

x = (0:0.1:10)'; y = sin(2*x)./exp(x/5); trnData = [x y];
trnData=[x y];
numMFs = 3;
mfType = 'gaussmf';
epoch_n = 3;
in_fisimat = genfis1(trnData,numMFs,mfType)
out_fisimat = anfis(trnData,in_fisimat,3);
writefis(out_fisimat,'sgaus7');
type sgaus7.fis
ew=evalfis(x,out_fisimat);
%save ew1.dat ew -ascii
plot(x,y,x,ew);
legend('Training Data','ANFIS Output')
%e=ew-y;plot(x,e);legend('chyba');pause
%sl=sum(e)
%sl2=sum(e.*e)
%sl3=sum(abs(e))
ruleview('sgaus7')
surfview('sgaus7')
fisimat1=readfis('sgaus7.fis');

```

### **Príloha č. 6 – program anfu.m pre tretí model**

```
x=tuy1(:,1);u=tuy1(:,2);y=tuy1(:,3);trnData=[x u y];%zo schemy sim5
numMFs = 5; %skus 5, lebo potom je 100 pravidiel
mfType = 'gaussmf';
epoch_n = 3;
in_fismat = genfis1(trnData,numMFs,mfType)
out_fismat = anfis(trnData,in_fismat,30);
writefis(out_fismat,'swgaus7a');
type swgaus7a.fis
ew=evalfis([x u],out_fismat);
% save ew1.dat ew -ascii
plot(x,y,x,ew);
legend('Training Data','ANFIS Output'),pause
%e=ew-y;plot(x,e);legend('chyba');pause
%s1=sum(e)
%s2=sum(e.*e)
%s3=sum(abs(e))
ruleview('swgaus7a'),pause
surfview('swgaus7a')
fismat=readfis('swgaus7a.fis');%pre simfu5
```