

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
KATEDRA ŘÍDÍCÍ TECHNIKY



**Modelování nelineárních systémů pomocí algoritmů shlukové
analýzy**
BAKALÁŘSKÁ PRÁCE

Prohlášení

Prohlašuji, že jsem svou diplomovou (bakalářskou) práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne

.....
Jiří Chuman

Poděkování

Rád bych na tomto místě poděkoval především mému vedoucímu bakalářské práce Ing. Petru Huškovi, Ph.D. za poskytnutí konzultací všech materiálů a konzultací v oblasti shlukové analýzy a fuzzy modelování a všem, bez kterých by tato práce nemohla vzniknout.

Abstrakt

Tato práce předkládá návrh modelování nelineárního systému pomocí algoritmů shlukové analýzy, Fuzzy c-means algoritmu a Gustafson-Kessel algoritmu. Dále je popsán způsob modelování pomocí grafického uživatelského rozhraní Fuzzy toolboxu pro MATLAB® a jeho propojení se simulačním prostředím SIMULINK™. V další části je popsán způsob vytvoření pravidel pro Mamdaniho model z projekce výsledných shluků. V závěru je porovnán vytvořený fuzzy model s původním nelineárním modelem a posouzena přesnost obou algoritmů.

Abstract

In this paper I will explore the design of simulation of a nonlinear system by fuzzy clustering algorithms, Fuzzy c-means algorithm and Gustafson-Kessel algorithm. In next part the way of simulation using Grafical User's Interface Fuzzy toolbox for MATLAB® is described. In next chapter the rules for Mamdani model from cluster projection are created and the results of created model are compared with the original nonlinear model and the accuracy of both algorithms is qualified.

Obsah

Abstrakt	1
1. Úvod	5
2. Volba systému a shluková analýza	5
2.1 Nelineární systém	6
2.2 Shluková analýza	7
2.3 Klasifikace algoritmů shlukové analýzy	7
2.4 Fuzzy shluková analýza	8
2.5 Fuzzy c-means algoritmus (FCM)	9
2.6 Gustafson-Kessel algoritmus (GK)	11
3. Modelování systému	14
3.1 Fuzzy systém	14
3.2 Fuzzy toolbox pro MATLAB	15
3.2.1 Grafické uživatelské rozhraní	15
3.2.2 Propojení s prostředím SIMULINK	20
3.3 Vytvoření Mamdani modelu pomocí Fuzzy toolboxu pro MATLAB	21
3.3.1 Projekce shluků	21
3.3.2 Tvorba pravidel	22
3.3.3 Propojení fuzzy modelu s prostředím SIMULINK	22
3.4 Porovnání fuzzy modelu a nelineárního modelu	23
4. Závěr	34
Literatura	35

Seznam obrázků

2.1	Vstupní signál	6
2.2	FCM shluky	10
2.3	Tvar a velikost GK shluku	11
3.4	Struktura fuzzy systému	15
3.5	FIS Editor	16
3.6	MF editor	17
3.7	Volba funkcí příslušnosti	18
3.8	Rule editor	18
3.9	Rule Viewer	19
3.10	Surface Viewer	20
3.11	Simulační schéma fuzzy modelu	23
3.12	Simulační schéma nelineárního modelu	23
3.13	Odezva na náhodný signál nelineárního modelu a fuzzy modelu - 3 pravidla	24
3.14	Odezva na sinusový signál s $\omega=2\pi/30$, $A=1$ nelineárního modelu a fuzzy modelu - 3 pravidla	25
3.15	Přechodová charakteristika nelineárního modelu a fuzzy modelu - 3 pravidla	26
3.16	Odezva na sinusový signál s $\omega=2\pi/30$, $A=1$ nelineárního modelu a fuzzy modelu - 7 pravidel	27
3.17	Přechodová charakteristika nelineárního modelu a fuzzy modelu - 7 pravidel	27
3.18	Odezva na sinusový signál s $\omega=2\pi/30$, $A=1$ nelineárního modelu a fuzzy modelu -10 pravidel	28
3.19	Přechodová charakteristika nelineárního modelu a fuzzy modelu - 10 pravidel	29
3.20	Odezva na sinusový signál s $\omega=2\pi/30$, $A=1$ nelineárního modelu a fuzzy modelu (1000 dat) -10 pravidel	30

3.21	Přechodová charakteristika nelineárního modelu a fuzzy modelu (1000 dat) -10 pravidel	31
3.22	Odezva nelineárního modelu a fuzzy modelu - 10 pravidel (naučeného z 1000 náhodných dat) na obdélníkový vstup se střídou 20% , T=50 vzorků, A=1	32
3.23	Odezva nelineárního modelu a fuzzy modelu - 10 pravidel (naučeného z 1000 náhodných dat) na obdélníkový vstup se střídou 50% , T=50 vzorků, A=1	32
3.24	Odezva nelineárního modelu a fuzzy modelu - 10 pravidel (naučeného z 1000 náhodných dat) na obdélníkový vstup se střídou 70% , T=50 vzorků, A=1	33

Seznam tabulek

1	Odchyly fuzzy modelů (RMSE) od nelineárního modelu	31
2	Chyba fuzzy modelů na obdélníkový vstupní signál T=50 vzorků, A=1	33

Kapitola 1

Úvod

S problémem rozpoznávání podobnosti mezi objekty, její kvantifikací a s tříděním objektů do skupin podle míry jejich podobnosti se v běžném životě setkáváme poměrně často a intuitivně jej bez problémů řešíme. Když chceme úlohy tohoto typu algoritmizovat a automatizovat, zjistíme, že problém je komplikovanější, než se jeví. V této souvislosti mluvíme o shlukové analýze. Aplikace použití jsou různorodé, např. při zpracování signálů, při rozpoznávání, v technické diagnostice, při řešení expertních a databázových systémů a při identifikaci modelů soustav.

Tato práce se zabývá modelováním nelineárního systému z naměřených dat pomocí dvou algoritmů shlukové analýzy, Fuzzy c-means algoritmu a Gustafson-Kessel algoritmu. Druhá kapitola popisuje vybraný nelineární systém a princip shlukové analýzy a obou zmíněných algoritmů. Další kapitola přináší popis modelování systému pomocí grafického uživatelského rozhraní Fuzzy toolboxu pro MATLAB® a jeho propojení se simulačním prostředím SIMULINK™. Je zde popsán návrh Mamdaniho modelu a princip projekce výsledných shluků na vytvoření pravidel pro fuzzy systém. V závěru je porovnám vytvořený fuzzy model s původním nelineárním modelem a posouzena přesnost obou algoritmů vzhledem k charakteru a množství dat pro učení.

Kapitola 2

Volba systému a shluková analýza

V této části popíšeme vybraný nelineární systém a vysvětlíme pojem shluková analýza a fuzzy shluková analýza. Dále ukážeme popis dvou algoritmů shlukové analýzy použitých k modelování.

2.1 Nelineární systém

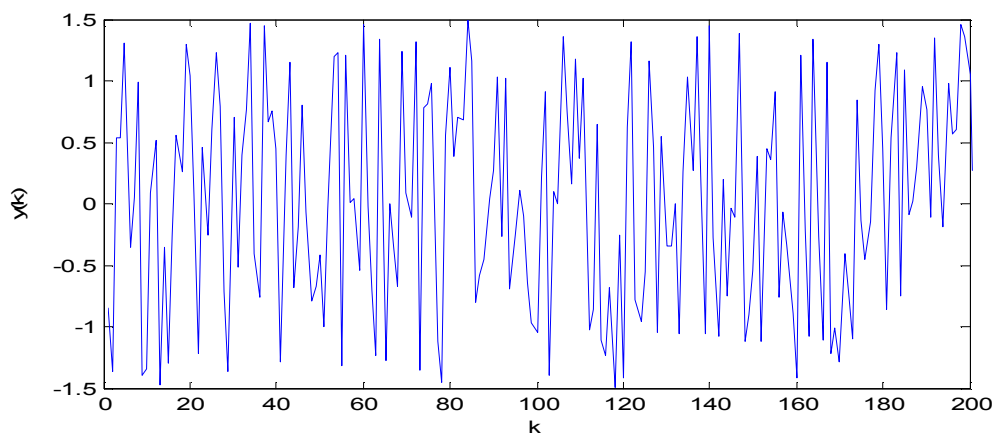
Pro návrh byl použit diskretní nelineární systém druhého řádu popsany rovnicí [6]:

$$y(k) = f(y(k-1), y(k-2)) + u(k), \quad (2.1)$$

kde

$$f(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)[y(k-1)-0.5]}{1 + y^2(k-1) + y^2(k-2)}. \quad (2.2)$$

Nelineární funkci f budeme aproximovat pomocí fuzzy modelu. Pro danou nelineární závislost se dvěma vstupy a jedním výstupem byla vygenerována simulační data pro náhodný vstupní signál s normálním rozložením v rozmezí hodnot $\langle -1.5, 1.5 \rangle$ viz obr. 2.1.



Obr. 2.1 Vstupní signál

2.2 Shluková analýza

Cílem shlukové analýzy je rozdělení objektů do shluků tak, aby byly splněny následující podmínky

- homogenita uvnitř shluku, t.j. objekty uvnitř shluku byly co nejpodobnější
- heterogenita mezi shluky, t.j. objekty mezi shluky byly co nejméně podobné.

Podobnost se v tomto případě převádí na vzdálenost. Nejpoužívanější typy výpočtů vzdálenosti jsou :

- Euklidovská vzdálenost $d^2(x_k, v_i) = (x_k - v_i)^T (x_k - v_i)$, (2.3)

- Manhattanská vzdálenost $d^2(x_k, v_i) = (x_k - v_i)^T \cdot A \cdot (x_k - v_i)$, (2.4)

- Mahalanobisova vzdálenost $d^2(x_k, v_i) = (x_k - v_i)^T \cdot F_i^{-1} \cdot (x_k - v_i)$, (2.5)

kde x_k jsou jednotlivá data, v_i je střed i -tého shluku,

$$A_i = \left[(\det(F_i))^{1/n} F_i^{-1} \right]^{1/n} \cdot F_i^{-1}, \quad (2.6)$$

kde F_i je kovarianční matice.

2.3 Klasifikace algoritmů shlukové analýzy

Nyní provedeme základní rozdělení algoritmů shlukové analýzy.

- 1) Heuristické shlukování – jedná se o projekční metodu, kde vícerozměrné objekty jsou převedeny na dvou nebo třírozměrné,
- 2) Deterministické shlukování – objekty přísluší právě jednomu shluku,
- 3) Shlukování s překryvem – objekty mohou patřit do několika shluků najednou,
- 4) Hierarchické shlukování - objekty jsou hierarchicky rozřazeny do shluků
- 5) Pravděpodobnostní shlukování - každému objektu je přiřazena pravděpodobnost,

udávající příslušnost ke shluku.

- 6) Shlukování využívající objektivní funkci – minimalizuje se chyba rozdělení shluků pomocí optimalizační funkce

Tyto algoritmy lze kombinovat a v této práci podrobněji probereme shlukování využívající objektivní funkci a zavedeme pojem fuzzy shluková analýza.

2.4 Fuzzy shluková analýza

U klasických metod shlukování je příslušnost jednotlivých dat do shluků popsána pomocí dvou hodnot $[0,1]$. Tyto hodnoty nám pouze říkají, jestli daný objekt do shluku náleží (hodnota 1) a nebo nenáleží (hodnota 0).

U fuzzy shlukování je příslušnost vyjádřena hodnotou z intervalu $\langle 0;1 \rangle$. To nám dává přesný popis pro shluky, které se vzájemně překrývají a popis pomocí dvou hodnot by nebyl možný.

Dále definujeme pojem objektivní funkce. Konkrétní podoba funkce záleží na použité technice shlukování. V našem případě volíme tvar funkce podle [5] takto :

$$J(f) = \sum_{x \in X} \sum_{k \in K} f^m(x)(k) \cdot d^2(x, k), \quad (2.7)$$

kde $f^m(x)(k)$ udává stupeň příslušnosti objektu $x \in X$ ke shluku $k \in K$ vztaženo ke všem shlukům, a $d(x, k)$ udává vzdálenost mezi objektem x a shlukem k .

Hodnota objektivní funkce nám udává chybu měření. Pomocí této funkce můžeme porovnávat různá řešení provedená na stejné množině dat. Cílem shlukové analýzy je minimalizovat tuto funkci.

Nyní si ukážeme základní strukturu algoritmu shlukové analýzy :

Na vstupu máme objekty ve tvaru $X = \{x_1, x_2, \dots, x_n\}$, tyto objekty představují námi změřená data na nelineárním systému.

1. krok – zvolíme počet shluků c tak, aby $2 \leq c < n$
2. krok – zvolíme parametr $m > 1$
3. krok – zvolíme hodnotu povolené chyby ε

Inicializujeme matici $U^{(0)}$ o velikosti $c \times n$, $l := 0$

REPEAT

Zvětšíme l o 1

Vypočteme množinu středů shluků $K^{(l)} = \{k_1, k_2, \dots, k_c\}$ tak, aby funkce (2.7)

byla minimální z $K^{(l)}$ pro $U^{(l-1)}$

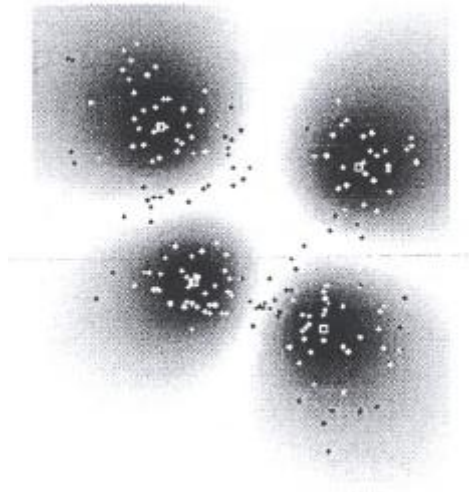
Vypočteme $U^{(l)}$, kde $u_{i,j} := f(x_j)(k_i)$

UNTIL $\|U^{(l-1)} - U^{(l)}\| \leq \varepsilon$

Podle tohoto schématu pracuje převážná část algoritmů fuzzy shlukové analýzy, pouze techniky výpočtu objektivní funkce a vzdálenosti se pro jednotlivé algoritmy liší a my si detailněji popíšeme dva základní algoritmy použité při návrhu nelineárního systému.

2.5 Fuzzy c-means algoritmus (FCM)

Jedná se o algoritmus, který pro výpočet vzdálenosti využívá Euklidovskou vzdálenost. Výsledné shluky mají tvar kružnice ve dvojdimenzionálním prostoru a shluky mají přibližně stejnou velikost. Každý shluk je reprezentován vlastním středem.



Obr. 2.2 FCM shluky

Na obr.2.2 je příslušnost ke shluku dána stupněm šedi, vyšší příslušnost tmavou barvou, malá příslušnost světlou barvou, a pro lepší názornost příslušnost menší jak $\frac{1}{2}$ není znázorněna.

Popis algoritmu :

Zvolíme počet shluků c tak, aby $2 \leq c < n$, parametr $m > 1$, $\varepsilon > 0$ a inicializujeme matici U náhodnými čísly,

REPEAT pro $l = 1, 2, \dots$

Krok 1 - vypočteme středy shluků :

$$v_i^{(l)} = \frac{\sum_{k=1}^n (u_{ik}^{(l-1)})^m x_k}{\sum_{k=1}^n (u_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c. \quad (2.8)$$

Krok 2 – vypočteme vzdálenost :

$$D_{ik}^2 = (x_k - v_i)^T (x_k - v_i), \quad 1 \leq i \leq c, \quad 1 \leq k \leq n. \quad (2.9)$$

Krok 3 – vypočteme novou matici U :

$$u_{i,k}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ik} / D_{jk})^{2/(m-1)}}. \quad (2.10)$$

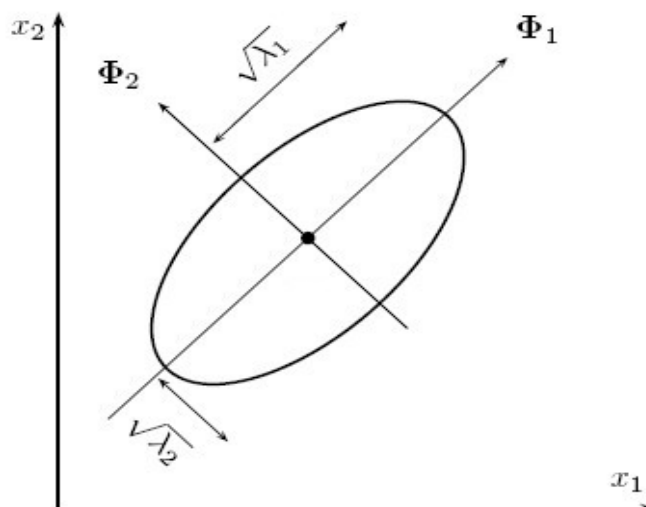
UNTIL $\|U^{(l-1)} - U^{(l)}\| \leq \varepsilon$

Experimenty bylo zjištěno, že nejlepších výsledků je dosaženo pro parametr $m=2$. Pokud bychom parametr m zvolili roven 1, pak matice U obsahuje pouze hodnoty [0;1] a z fuzzy algoritmu se stává nefuzzy hard c-means algoritmus.

Výhoda fuzzy c-means algoritmu je jeho malá časová náročnost.

2.6 Gustafson-Kessel algoritmus (GK)

Na rozdíl od předchozího algoritmu, GK algoritmus používá pro výpočet Manhattanskou vzdálenost. Každý shluk je charakterizován symetrickou a pozitivně definitní maticí A . Shluky mají tvar elipsy ve dvourozměrném prostoru a mají konstantní velikost, z čehož vyplývá, že determinant matice A je konstantní. Délka poloos elipsy a umístění shluku jsou dány vlastními čísly a vlastními vektory matice A , tak jak je znázorněno na obr. 2.3.



Obr. 2.3 Tvar a velikost GK shluku

Popis algoritmu :

Počáteční zvolení parametrů je stejné jako v předchozím případě.

REPEAT pro $l = 1, 2, \dots$

Krok 1 - vypočteme středy shluků :

$$v_i^{(l)} = \frac{\sum_{k=1}^n (u_{ik}^{(l-1)})^m x_k}{\sum_{k=1}^n (u_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c. \quad (2.11)$$

Krok 2 – vypočteme kovarianční matici :

$$F_i^{(l)} = \frac{\sum_{k=1}^n (u_{ik}^{(l-1)})^m (x_k - v_i^{(l)})(x_k - v_i^{(l)})^T}{\sum_{k=1}^n (u_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c. \quad (2.12)$$

- určíme vlastní čísla λ_{ij} a vlastní vektory ϕ_{ij} matice $F_i^{(l)}$, které nám dávají informaci o poloze shluku

Krok 3 – vypočteme vzdálenost :

$$D_{ikA_i}^2 = (x_k - v_i^{(l)})^T \left[(\det(F_i^{(l)}))^{1/n} (F_i^{(l)})^{-1} \right] (x_k - v_i^{(l)}) \quad (2.13)$$

Krok 4 – vypočteme novou matici U :

$$u_{i,k}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ik} / D_{jk})^{2/(m-1)}}, 1 \leq i \leq c, 1 \leq k \leq n. \quad (2.14)$$

UNTIL $\|U^{(l-1)} - U^{(l)}\| \leq \varepsilon$.

V porovnání s fuzzy c-means algoritmem je časová náročnost GK algoritmu vyšší, zejména pro vícerozměrná data. Počet cyklů vedoucí k výsledku je v porovnání s předchozím algoritmem menší.

Kapitola 3

Modelování systému

U obou dvou algoritmů dostáváme jako výsledek informaci o shlucích, včetně jejich středů, a na základě toho můžeme přikročit k modelování fuzzy systému. Modelování je založeno na popisu shluků pomocí fuzzy pravidel. Jeden z možných nástrojů nám poskytuje Fuzzy toolbox pro MATLAB[®] a jeho propojení s grafickým editorem SIMULINK[™].

3.1 Fuzzy systém

Pod pojmem *fuzzy systém* rozumíme takový systém, jehož vstupní a výstupní proměnné nabývají hodnot (stavů), které nejsou definovány ostrými čísly, ale jsou definovány slovními hodnotami, tj. fuzzy množinami. Jedná se tedy o jazykové (lingvistické) proměnné, které jsou definovány na množině (univerzu). Velmi často bývá touto množinou interval reálných čísel, kde každému prvku univerza přiřazujeme stupeň příslušnosti do jedné nebo více fuzzy množin odpovídajících významu základních termů použitých v pravidlech. Rozkladem celého univerza dostaneme *funkci příslušnosti*, jejíž tvar může být různý a ve většině případech jsou z důvodu snadnosti výpočtu voleny tvary sestavené z lomených přímek (trojúhelníková Λ – funkce, lichoběžníková Π - funkce, L – funkce, Γ - funkce atd.), ale používají se i jiné tvary funkcí příslušnosti, jako např. Gaussova funkce, Zadehova S-funkce atd.

Další velice důležitou částí fuzzy systému je *inferenční mechanismus*, který na základě vyhodnocení všech *fuzzy pravidel* nalezne výsledek přibližného úsudku. Jedná se o postup, který na základě vstupní fuzzy množiny, resp. ostré hodnoty stanoví výstupní fuzzy množinu při zvoleném způsobu interpretace implikace v pravidlech. Podle toho rozeznáváme dva typy fuzzy systémů.

Fuzzy systém typu Mamdani

Jedná se o fuzzy systém, kde v konsekventu pravidel je fuzzy výrok, neboli výstupní univerzum je pokryto souborem fuzzy množin. Pravidla mají tvar

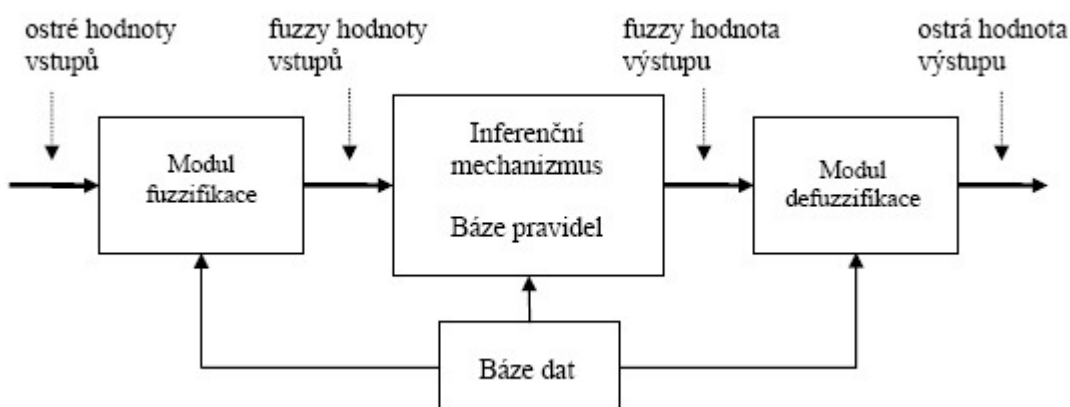
$$\text{If } \langle \text{fuzzy výrok} \rangle \text{ Then } \langle \text{fuzzy výrok} \rangle.$$

Fuzzy systém typu Takagi-Sugeno

Druhým typem fuzzy systému je systém, kde v konsekventu pravidel není fuzzy výrok, ale nějaká funkce vstupních proměnných. Pravidla mají tvar

If $\langle \text{fuzzy výrok} \rangle$ *Then* $f_k(x_1, \dots, x_n)$, kde funkce f_k jsou často lineární kombinací vstupních proměnných, $f_k(x_1, \dots, x_n) = a_{k0} + a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n$.

Ve většině technických aplikacích dostáváme vstupní proměnné ve formě ostrých naměřených údajů (reálných čísel) a také výstupní hodnoty jsou požadovány ve formě ostrých čísel. Pak je nutné doplnit fuzzy systém o modul *fuzzifikace* a modul *defuzzifikace*. Modul *fuzzifikace* zajišťuje převod ostré hodnoty na fuzzy množinu. Naopak úlohou modulu *defuzzifikace* je získat ostrou hodnotu z dané fuzzy množiny. Struktura fuzzy systému je ukázána na obr. 3.4



Orb. 3.4 Struktura fuzzy systému

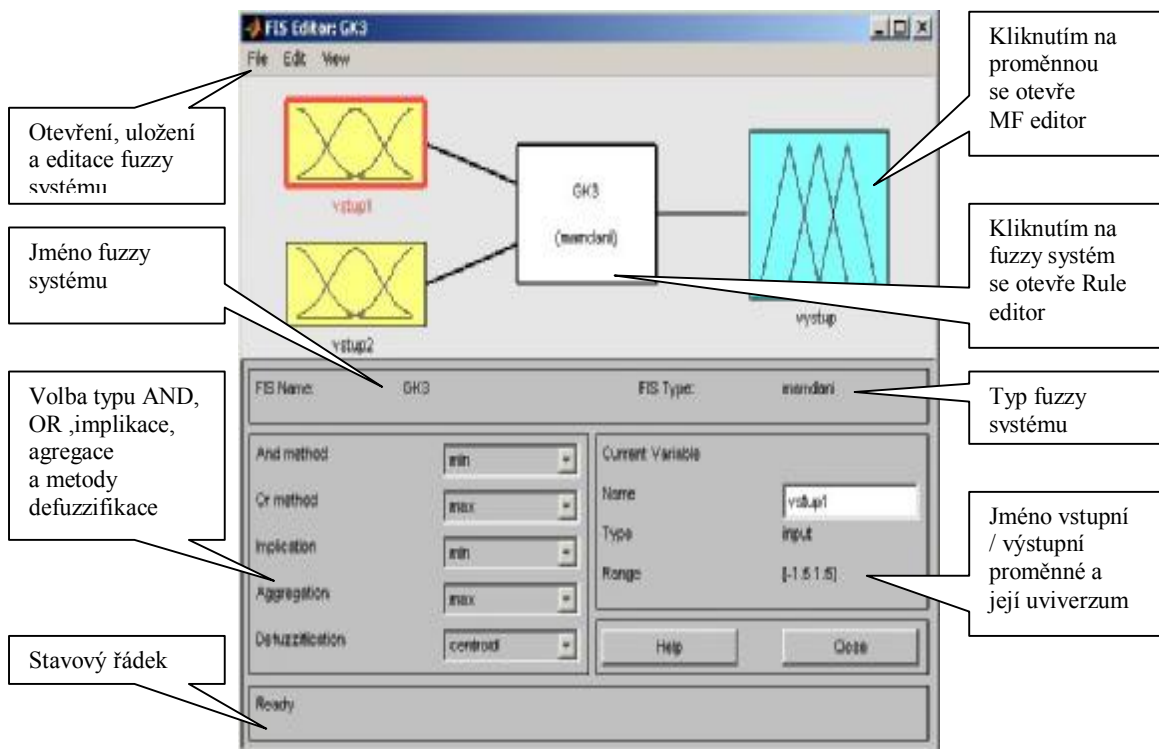
3.2 Fuzzy toolbox pro MATLAB

3.2.1 Grafické uživatelské rozhraní

Fuzzy toolbox je jedním z mnoha nástrojů simulačního prostředí MATLAB od firmy MathWorks. Umožňuje pomocí grafického rozhraní naplnit znalostní i datovou bázi fuzzy systému. Samozřejmě, jako v celém prostředí MATLAB, je možné pracovat s fuzzy systémem v příkazovém řádku. Z důvodu lepší názornosti se zaměříme právě na zmiňované grafické rozhraní, někdy též nazývané jako grafický uživatelský interface GUI (Graphical User Interface - GUI). Spuštění Fuzzy toolboxu se provede zadáním příkazu *fuzzy* v příkazovém řádku programu MATLAB.

Fuzzy toolbox se skládá ze tří nástrojů pro tvorbu a editaci fuzzy systému (FIS Editor, MF Editor a Rule Editor) a dvou pasivních prohlížečů (Rule Viewer a Surface Viewer).

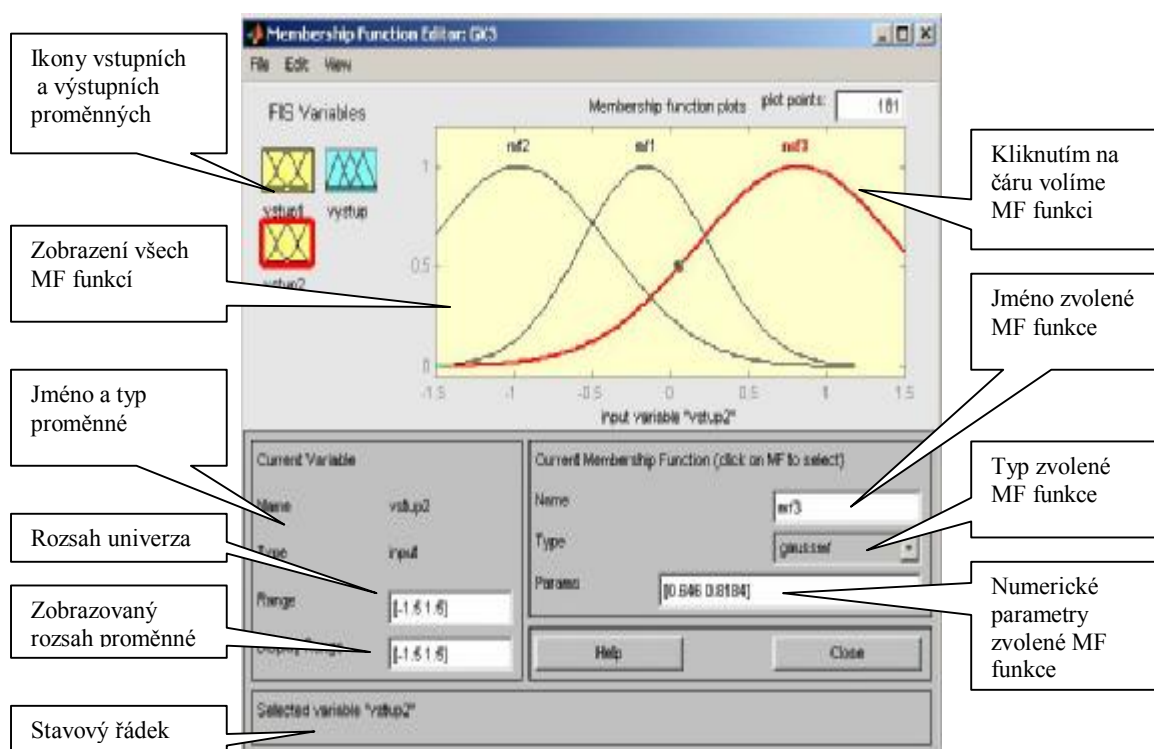
Hlavní částí toolboxu je **FIS Editor** (Fuzzy Inference System Editor) viz obr. 3.5, ve kterém se definují základní parametry systému, jako je počet vstupních a výstupních proměnných, jejich jména, metody fuzziifikace, defuzziifikace atd.



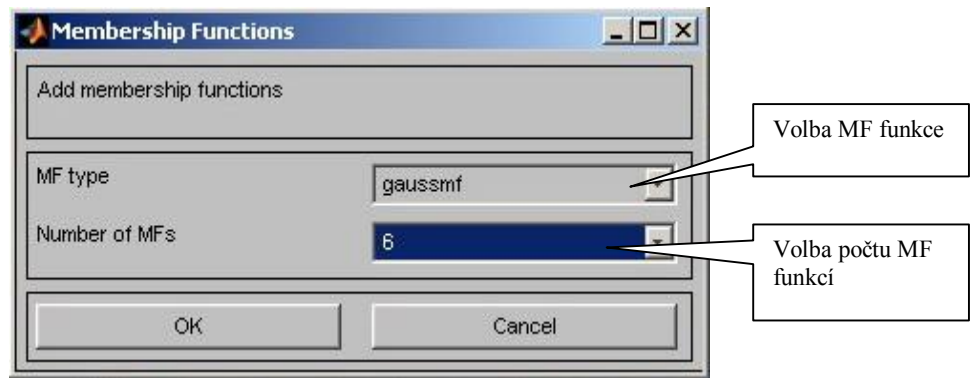
Obr. 3.5 FIS Editor

Dalším aktivním nástrojem je **MF Editor** (Membership Function Editor) viz obr. 3.6, pomocí kterého definujeme počet a tvar jednotlivých funkcí příslušnosti vstupních a výstupních proměnných a dále jejich jména a rozsahy. V okně „Range“ nastavíme odpovídající univerzum. V položce „Edit – Add MFs...“ volíme počet a tvar funkcí příslušnosti viz obr. 3.7. Předefinovány jsou následující tvary funkcí příslušnosti: trimf – trojúhelníkový, trapmf – trapezoidální, gbellmf – obecná zvonová křivka, gaussmf – Gaussova křivka, gauss2mf – dvoustranná Gaussova křivka, pimf – splajnová křivka tvaru Π , dsigmf - rozdíl dvou sigmoidálních křivek, který dává jako výsledek křivku tvaru Π , psigmf – součin dvou sigmoidálních křivek, který dává opět křivku tvaru Π .

Po navolení počtu a tvaru funkcí příslušnosti vybereme postupně jednotlivé funkce příslušnosti (kliknutím na jejich zobrazovaný průběh) a zadáme jejich jména. Můžeme též individuálně změnit jejich typ, popřípadě jejich numerické parametry (polohu v univerzu). Takto zadáme všechny dříve definované vstupní a výstupní proměnné.

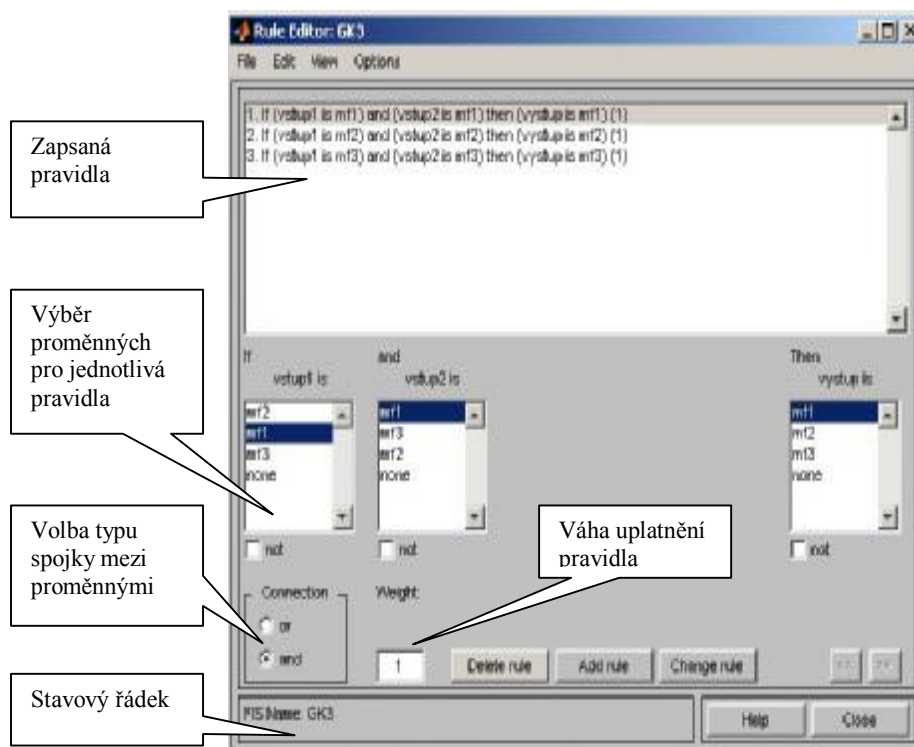


Obr. 3.6 MF Editor



Obr. 3.7 Volba funkcí příslušnosti

K zadávání znalostní báze fuzzy systému slouží **Rule Editor** viz obr. 3.8, který zobrazuje vstupní a výstupní proměnné nedefinované pomocí MF Editoru a umožňuje jednoduché vytvoření pravidel typu „IF – THEN“.

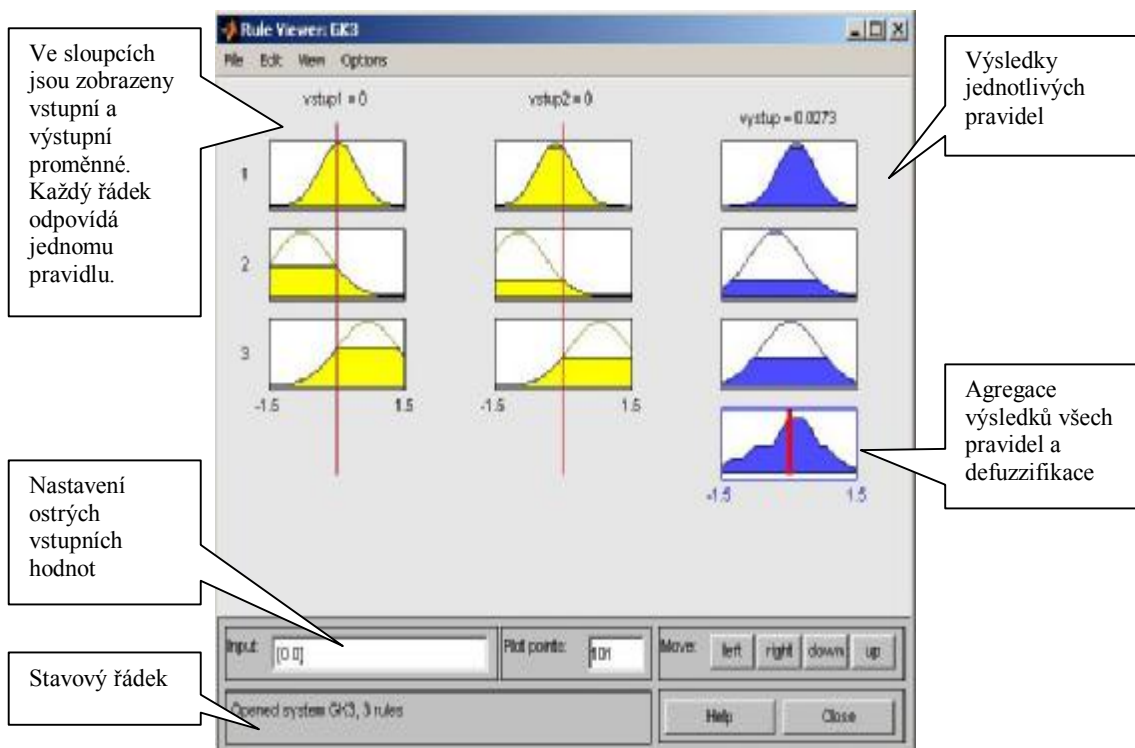


Obr. 3.8 Rule Editor

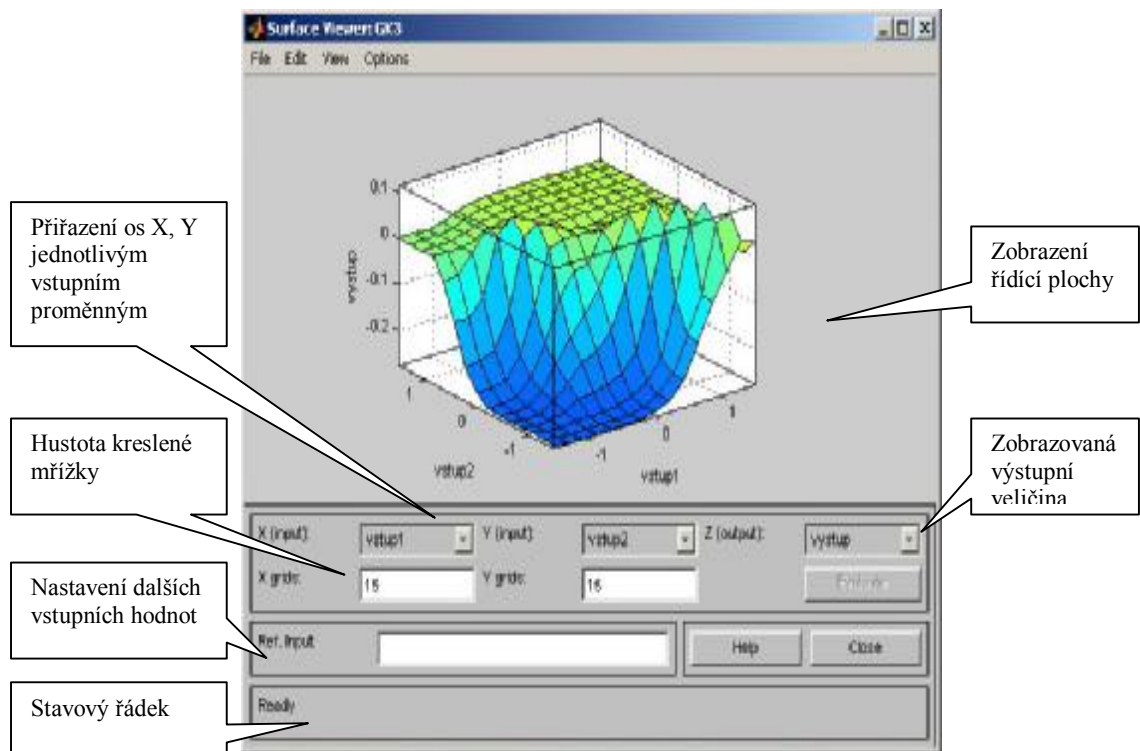
Poslední dva nástroje Rule Viewer obr. 3.9 a Surface Viewer obr. 3.10 slouží ke sledování chování navrhovaného systému. **Rule Viewer** umožňuje sledovat chování

systemu při konkrétních hodnotách vstupních proměnných. V řádcích jsou zobrazena pravidla, ve sloupcích pak vstupní a výstupní proměnné. Kurzory ve sloupcích vyznačují ostrou hodnotu vstupů a výstupu a ukazují, jakou měrou se na výstupních hodnotách systému podílejí jednotlivé vstupy. Ostré hodnoty lze též numericky zadat v editačním poli.

Druhým ladícím nástrojem Fuzzy toolboxu je **Surface Viewer** zobrazující řídicí plochu navrhovaného systému, která je funkcí n proměnných. Řídicí plocha je dána průchodem ostrých hodnot přes svá univerza, takže pro každou n -tici ostrých hodnot je provedena posloupnost činností fuzzifikace – inferenční mechanismus – defuzzifikace, což má za následek jednoznačné určení funkce n proměnných.



Obr. 3.9 Rule Viewer



Obr. 3.10 Surface Viewer

Výstupem navrhovaného systému je FIS matice nebo soubor s příponou *.fis, kde **FIS matice** je speciální MATLAB objekt obsahující všechny informace o navrhovaném fuzzy systému. V případě ukládání FIS matice na disk existuje speciální textový formát, ve kterém jsou údaje FIS matice uloženy.

3.2.2 Propojení s prostředím SIMULINK

Pro práci s fuzzy systémem v prostředí Simulink existují dva bloky umožňující načtení FIS matice či souboru *.fis. Jedná se o „Fuzzy Logic Controller“ a „Fuzzy Logic Controller with Ruleviewer“, který navíc umožňuje sledování průběhu pravidel modelovaného systému. Jak jsme již uvedli dříve, FIS matice, resp. soubor *.fis obsahují všechny potřebné informace o navrhovaném systému. Přidáním dalších simulačních bloků můžeme dotvořit modelovaný systém dle požadovaných vlastností.

3.3 Vytvoření Mamdani modelu pomocí Fuzzy toolboxu pro MATLAB

Při modelování systému musíme nejdříve provést výběr tvaru funkce příslušnosti. V našem případě jsme zvolili Gaussovu křivku, která je charakterizována rovnicí :

$$f(x, \sigma, k) = e^{\frac{-(x-k)^2}{2\sigma^2}}, \quad (3.1)$$

kde k je střed Gaussovy křivky (v našem případě střed shluku) a σ je její šířka. U GK algoritmu šířku funkce příslušnosti určují vlastní čísla λ_{ij} podle obr. 2.3, pouze u FCM algoritmu je situace o něco komplikovanější, protože na výstupu nedostáváme přímou informaci o tom, jak je který shluk velký a zde je nutno provést tzv. projekci shluků.

3.3.1 Projekce shluků

Pro popis fuzzy modelu je nutné popsat shluky pomocí jednotlivých fuzzy pravidel. Každé pravidlo reprezentuje jeden shluk. Střed funkce příslušnosti je dán středem shluku a šířku funkce nám udává rozměr shluku. Zde je nutné provést projekci.

Existuje více možných řešení, jak danou projekci provést. Lze použít grafické metody, ale vzhledem k tomu, že množina řešení se nalézá v n -dimenzionálním prostoru, není tato metoda nejvhodnější a museli bychom provádět složité transformace.

Lepší řešení nabízí použití numerické metody. Při použití těchto metod používáme při výpočtech jisté zjednodušení a tedy výsledná projekce zanáší do výsledku chybu, kterou se snažíme minimalizovat zvolením vhodné aproximace.

V našem případě jsme vyšli z výpočtu objektivní funkce (2.7) a to konkrétně z výpočtu stupně příslušnosti :

$$f(x)(k) = \frac{1}{\sum_{j \in K} \frac{d^2(x, k)}{d^2(x, j)}}, \quad (3.2)$$

kde K značí množinu středů shluků.

Pro náš systém definovaný dvěma vstupy a jedním výstupem je střed shluku definován takto : $k = (k_x, k_y, k_z)$.

Při volbě $x = k_x + 2\sigma$ a z rovnosti rovnic (3.1) a (3.2) dostáváme soustavu :

$$\frac{\sum_{j \in K} (k_x + 2\sigma - j_x)^2 + (k_y - j_y)^2 + (k_z - j_z)^2}{4\sigma^2} = e^{-2}, \quad (3.3)$$

jejímž vyřešením dostáváme parametr σ t.j. šířku funkce příslušnosti. Výpočet je nutno provést pro všechny tři souřadnice odděleně tak, že volíme za x vždy postupně jednu ze tří souřadnic středu. Hodnota funkce příslušnosti e^{-2} odpovídá vzdálenosti 2σ , t.j. $\sigma \sim e^{-\frac{1}{2}} \cong 0.6$, což je hodnota příslušnosti v daném bodě ke shluku k . To nám zaručuje, že tento bod náleží k danému shluku s dostatečně velkou pravděpodobností.

3.3.2 Tvorba pravidel

V předchozí části jsme si uvedli, jak je systém typu Mamdani definován. Fuzzy výroky jsou reprezentovány jednotlivými shluky t.j. funkcemi příslušnosti.

Jako příklad uveďme pravidla pro systém se třemi shluky

If $y(k-1)$ je shluk 1 And $y(k-2)$ je shluk 1 Then $f(y(k-1), (k-2))$ je shluk 1, **PRAVIDLO 1**

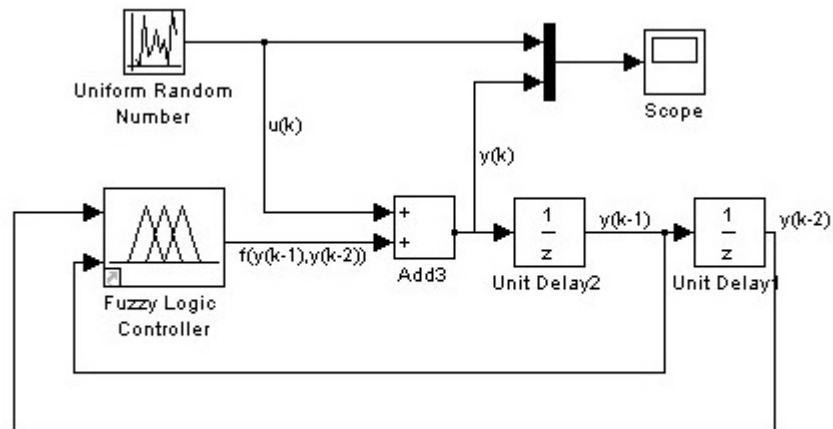
If $y(k-1)$ je shluk 2 And $y(k-2)$ je shluk 2 Then $f(y(k-1), (k-2))$ je shluk 2, **PRAVIDLO 2**

If $y(k-1)$ je shluk 3 And $y(k-2)$ je shluk 3 Then $f(y(k-1), (k-2))$ je shluk 3, **PRAVIDLO 3**

Nyní již máme všechny informace reprezentující chování Mamdani fuzzy modelu a můžeme vytvořenou FIS matici propojit se simulačním prostředím SIMULINK.

3.3.3 Propojení fuzzy modelu s prostředím SIMULINK

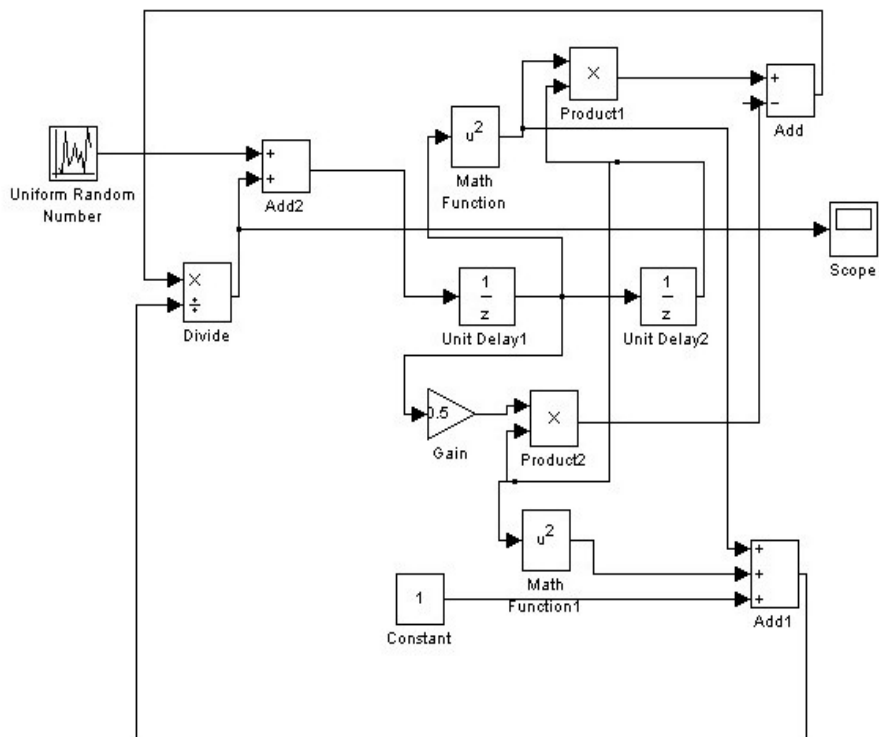
Hlavním blokem „Fuzzy Logic Controller“ načteme FIS matici a fuzzy model doplníme dalšími bloky tak, aby jsme mohli simulačně prověřit činnost systému v uzavřené smyčce.



Obr. 3.11 Simulační schéma fuzzy modelu

3.4 Porovnání fuzzy modelu a nelineárního modelu

Po vytvoření fuzzy modelu nyní můžeme porovnat model s původním nelineárním modelem, navrženým v simulačním prostředí SIMULINK viz obr. 3.12. Porovnávat budeme výsledné modely vytvořené pomocí FCM algoritmu a GK algoritmu.



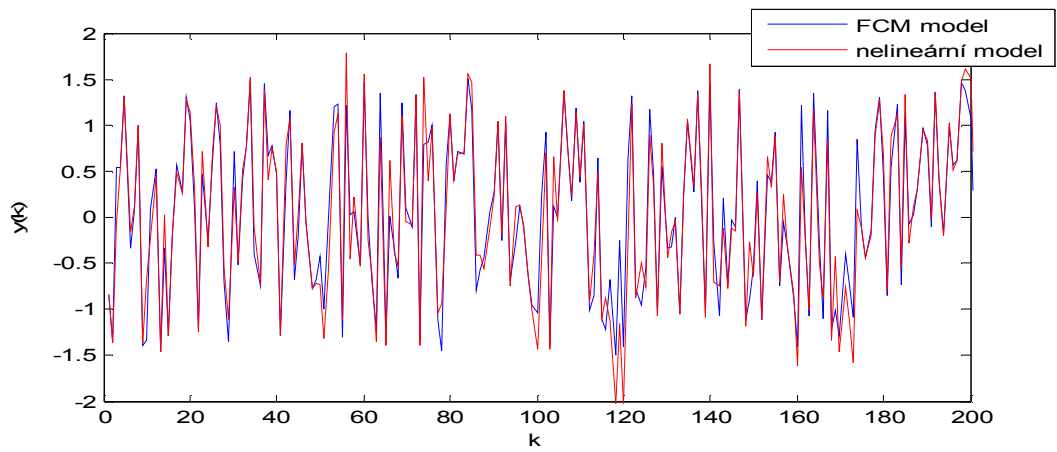
Obr. 3.12 Simulační schéma nelineárního modelu

Pro porovnání výsledků zavedeme střední kvadratickou odchylku RMSE podle vztahu :

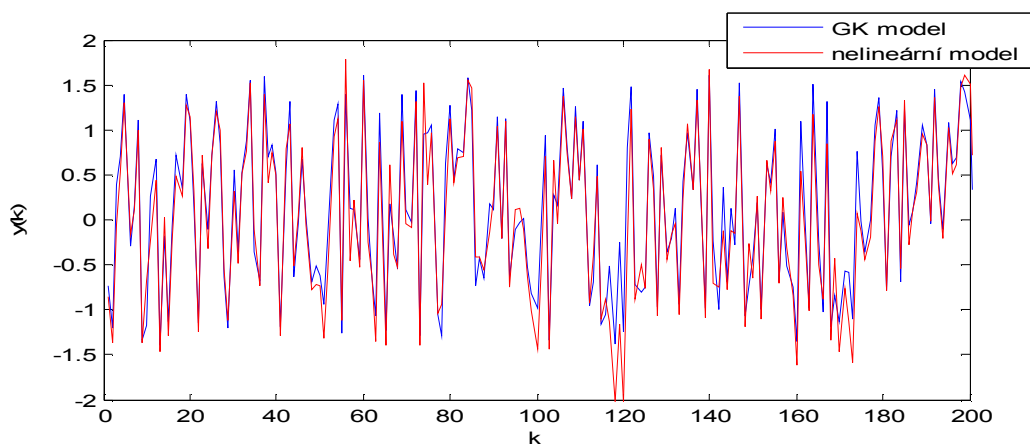
$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2}$$

Nejdříve porovnáme modely naučené z 200 naměřených dat a prozkoumáme vliv počtu pravidel (t.j. shluků) na výslednou odezvu modelu.

Odezvu budeme porovnávat pro náhodná vstupní data, pro vstupní sinusový signál s $\omega=2\pi/30$, $A=1$ a odezvu na jednotkový skok.

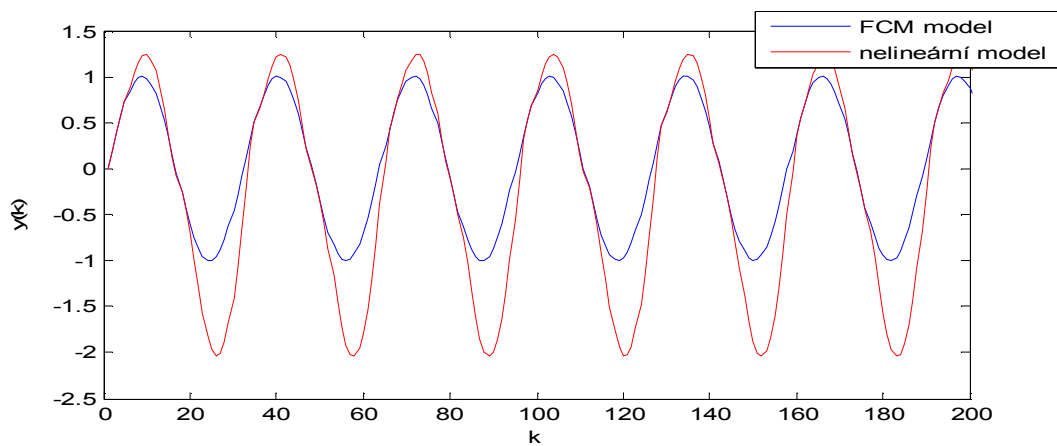


RMSE = 0.2642

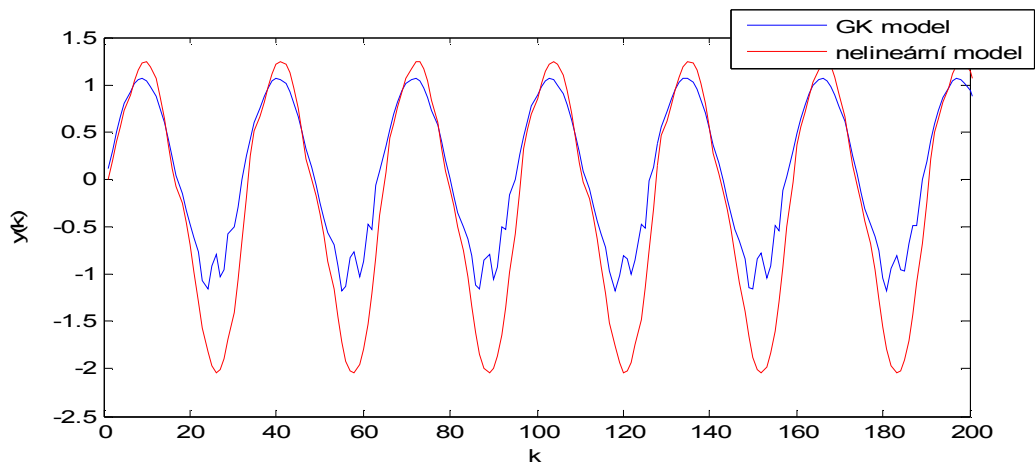


RMSE = 0.2581

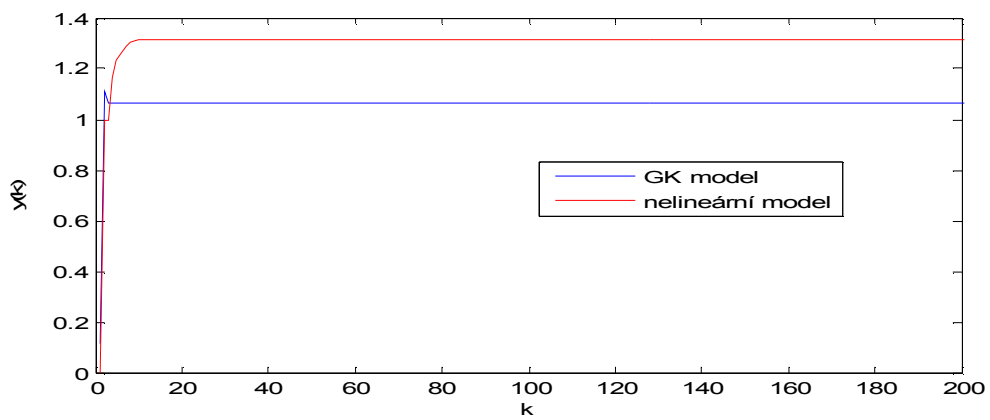
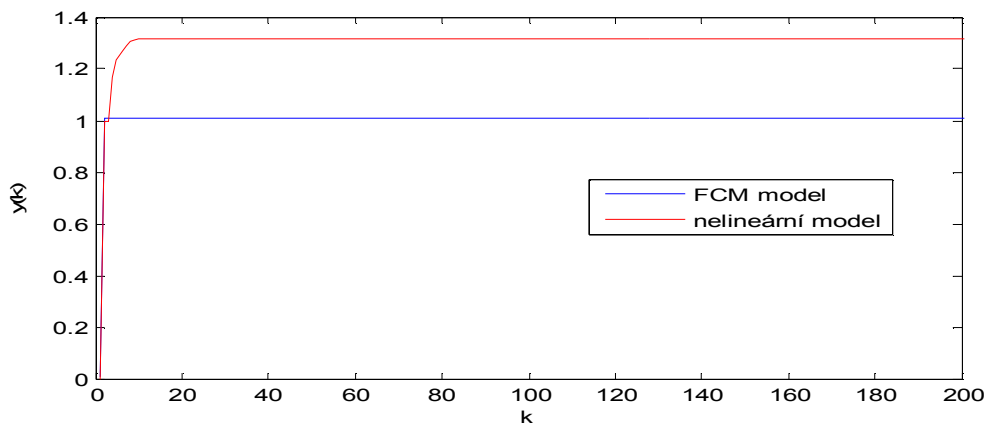
Obr. 3.13 Odezva na náhodný signál nelineárního modelu a fuzzy modelu - 3 pravidla



RMSE = 0.5296



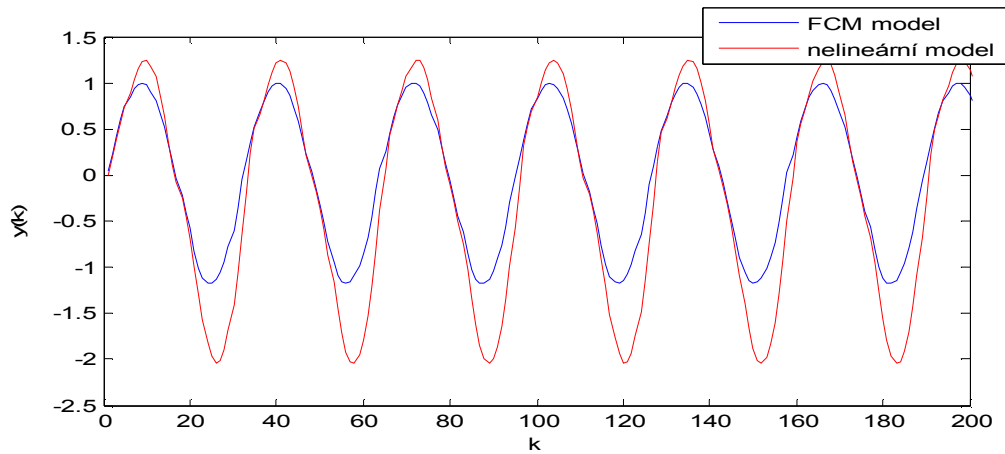
Obr. 3.14 Odezva na vstupní sinusový signál s $\omega=2\pi/30$, $A=1$ nelineárního modelu a fuzzy modelu - 3 pravidla



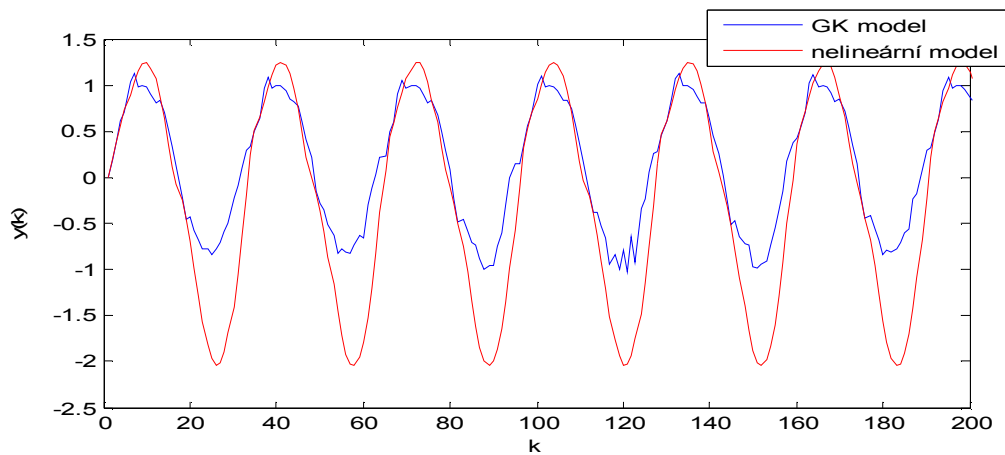
Obr. 3.15 Přejchodová charakteristika nelineárního modelu a fuzzy modelu - 3 pravidla

Pro odezvu na náhodný signál nelineárního modelu a fuzzy modelu - 7 pravidel jsou velikosti odchylek následující :

- pro FCM model **RMSE = 0.2343**
- pro GK model **RMSE = 0.3110**

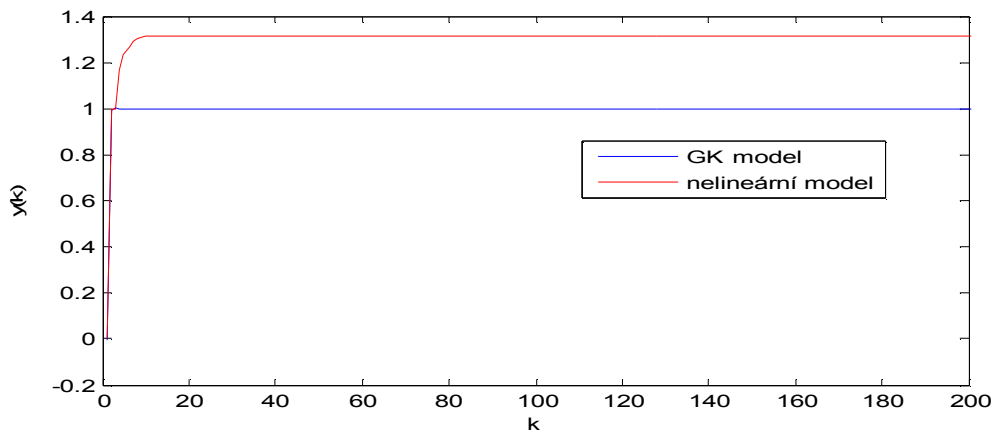
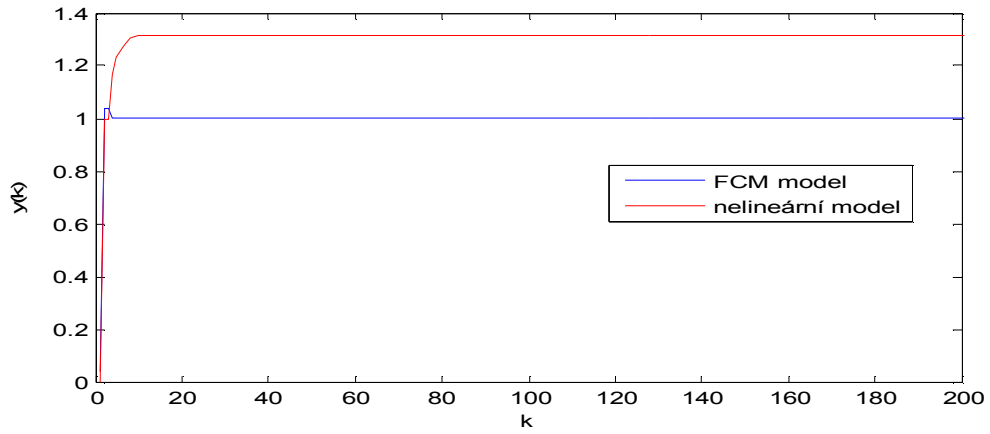


RMSE = 0.4543



RMSE = 0.6042

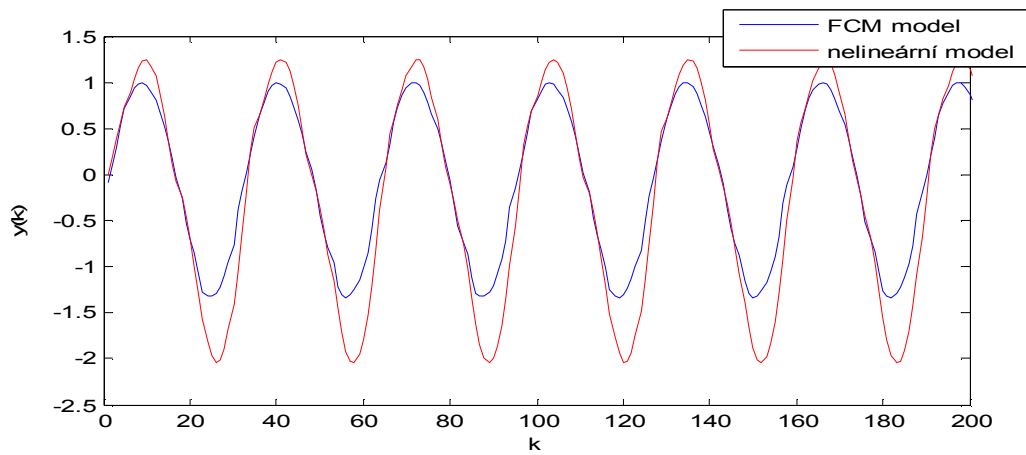
Obr. 3.16 Odezva na vstupní sinusový signál s $\omega=2\pi/30$, $A=1$ nelineárního modelu a fuzzy modelu -7 pravidel



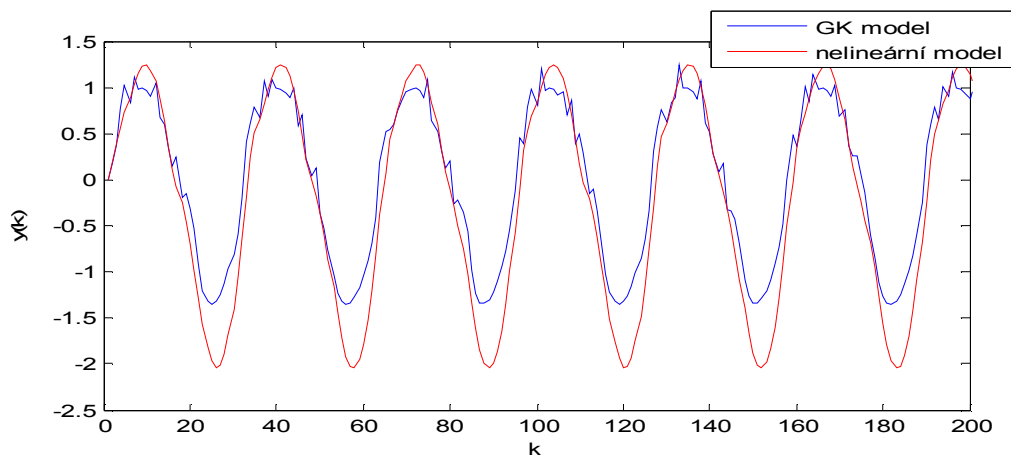
Obr. 3.17 Přechodová charakteristika nelineárního modelu a fuzzy modelu - 7 pravidel

Pro odezvu na náhodný signál nelineárního modelu a fuzzy modelu - 10 pravidel jsou velikosti odchylek následující :

- pro FCM model **RMSE = 0.2193**
- pro GK model **RMSE = 0.3012**

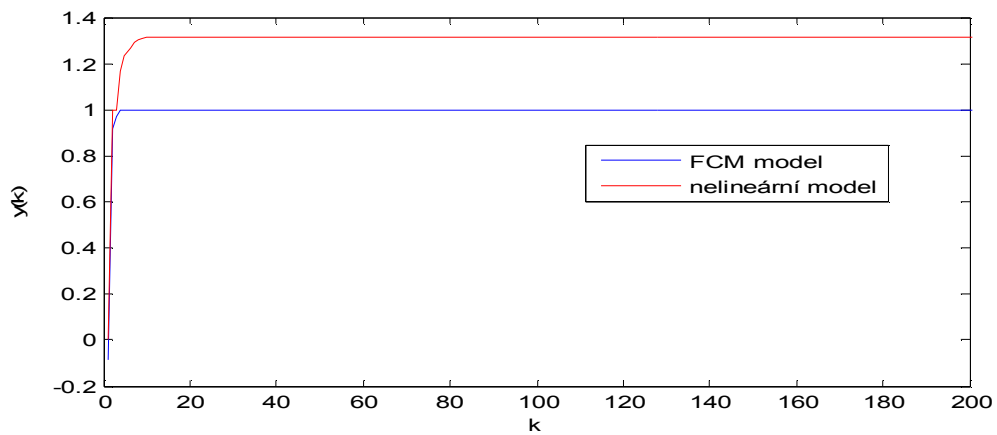


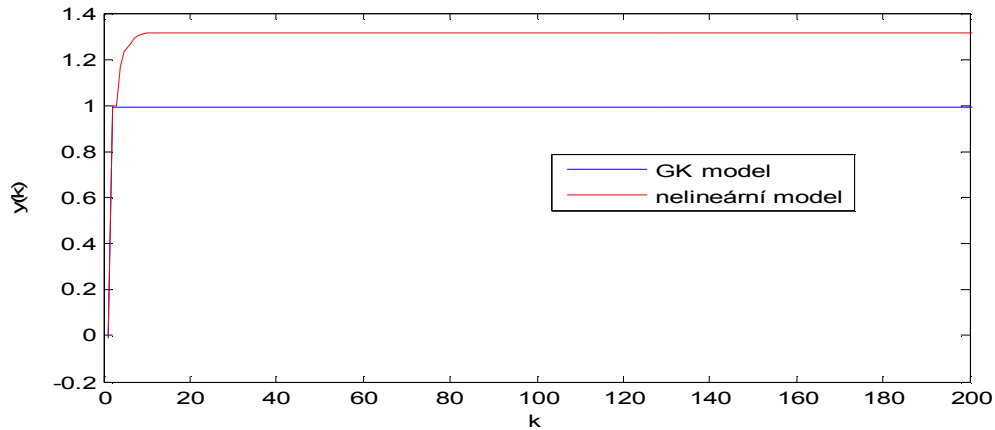
RMSE = 0.3755



RMSE = 0.3914

Obr. 3.18 Odezva vstupní sinusový signál s $\omega=2\pi/30$, $A=1$ nelineárního modelu a fuzzy modelu -10 pravidel





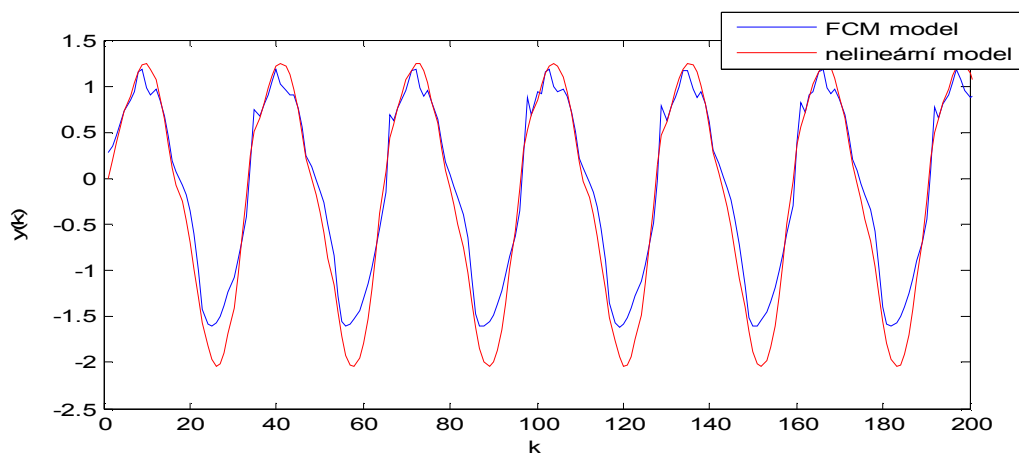
Obr. 3.19 Přechodová charakteristika nelineárního modelu a fuzzy modelu - 10 pravidel

Další zvyšování pravidel nevedlo ke snížení chyby, naopak docházelo k redundanci pravidel, kde se vzájemně jednotlivá pravidla překrývala a to vnáší další chybu do výsledků a ztrácí se přehlednost při návrhu.

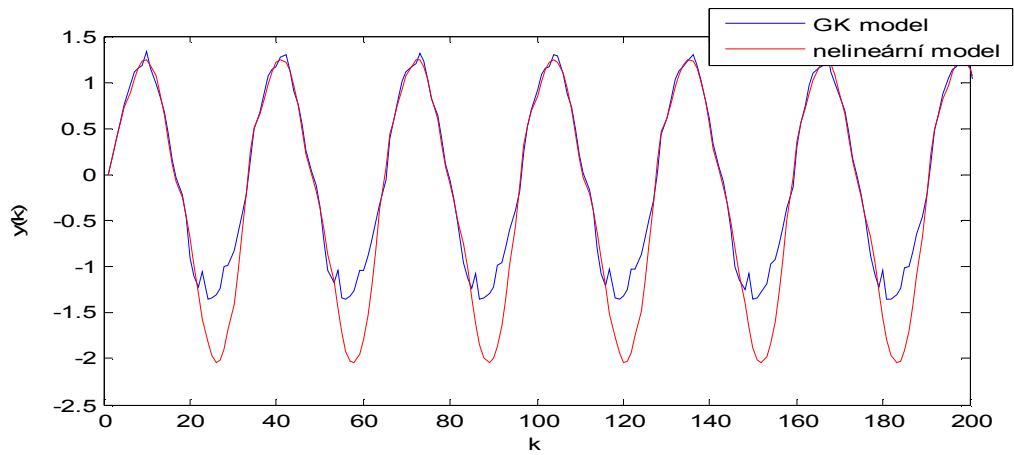
Jako další řešení zlepšení výsledků je možnost zvýšení počtu dat pro učení. Nyní provedeme návrh pro 1000 naměřených dat .

Pro odezvu na náhodný signál nelineárního modelu a fuzzy modelu - 10 pravidel jsou velikosti odchylek následující :

- pro FCM model **RMSE = 0.1929**
- pro GK model **RMSE = 0.2126**

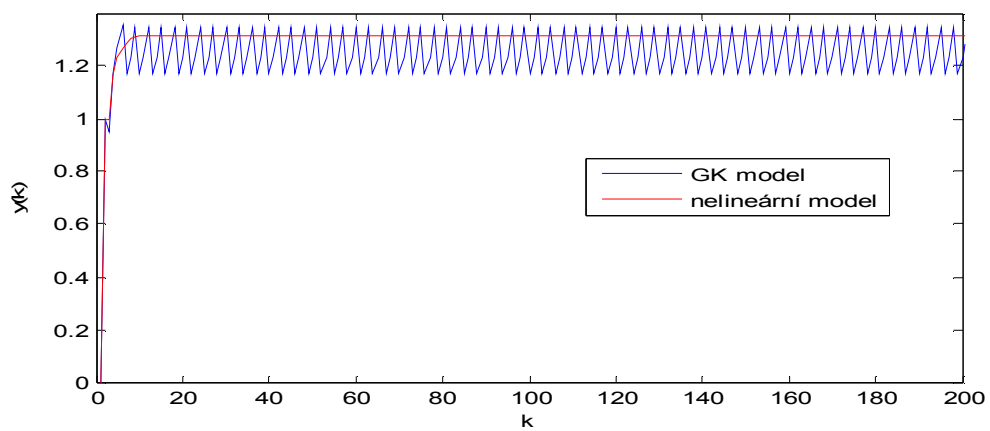
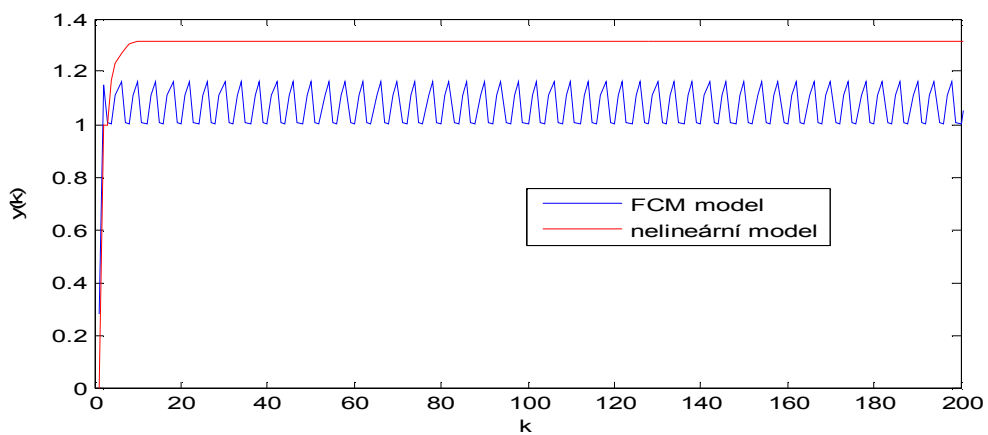


RMSE = 0.2538



RMSE = 0.3458

Obr. 3.20 Odezva na vstupní sinusový signál s $\omega=2\pi/30$, $A=1$ nelineárního modelu a fuzzy modelu (1000 dat) -10 pravidel



Obr. 3.21 Přechodová charakteristika nelineárního modelu a fuzzy modelu (1000 dat) - 10 pravidel

V následující tabulce shrneme dosažené výsledky v podobě odchylky fuzzy modelů od nelineárního modelu.

	systém naučený pro 200 dat					
	3 pravidla		7 pravidel		10 pravidel	
vstupní signál	FCM	GK	FCM	GK	FCM	GK
náhodný signál	0.2642	0.2581	0.2343	0.3110	0.2193	0.3012
<i>sin k</i>	0.5296	0.5218	0.4543	0.6042	0.3755	0.3914

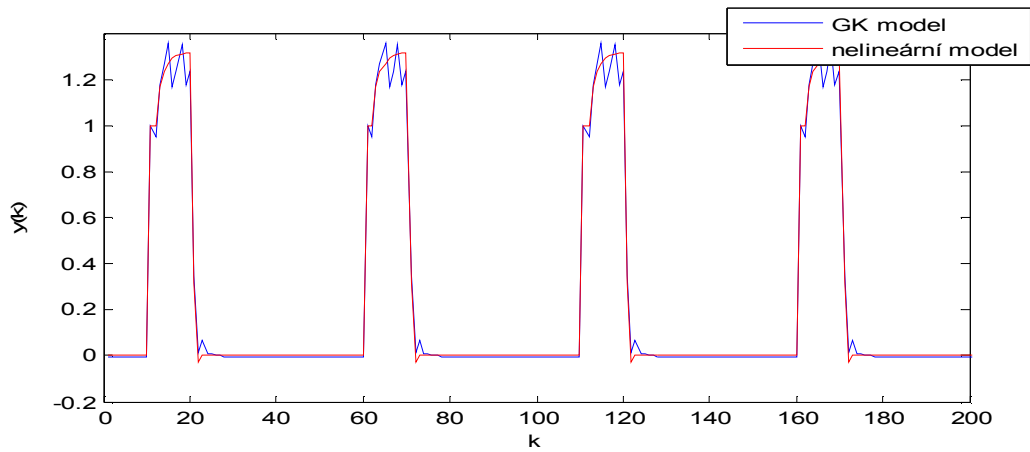
	systém naučený pro 1000 dat					
	3 pravidla		7 pravidel		10 pravidel	
vstupní signál	FCM	GK	FCM	GK	FCM	GK
náhodný signál	0.2722	0.2536	0.2200	0.2992	0.1929	0.2126
<i>sin k</i>	0.5704	0.5413	0.2966	0.3929	0.2538	0.3458

Tab. 1 Odchylky fuzzy modelů (RMSE) od nelineárního modelu

Zvýšením počtu dat pro učení došlo ke zpřesnění návrhu pomocí GK algoritmu, ale zároveň došlo k rozkmitání celého systému, jak ukazují výsledné přechodové charakteristiky.

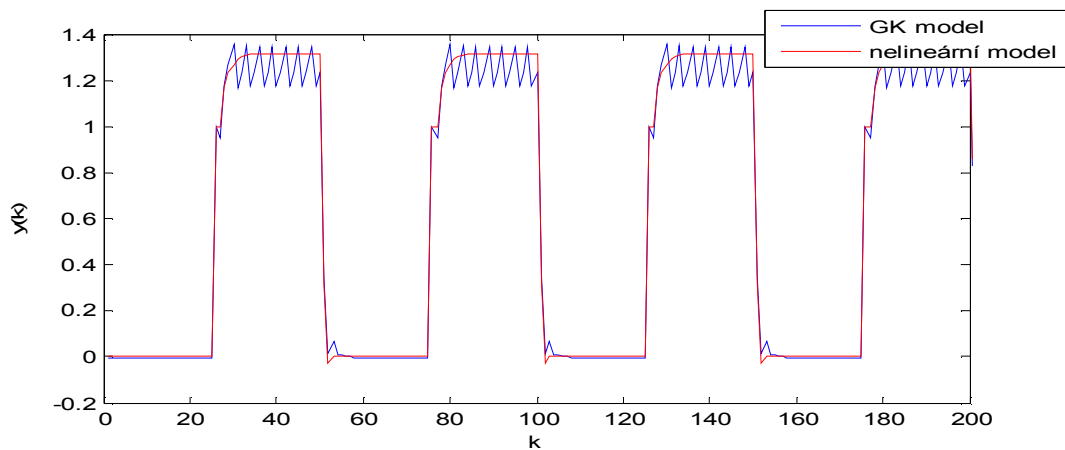
Další zvětšení počtu dat pro učení nepřineslo zlepšení výsledného návrhu, to je zapříčiněno tím, že ve vstupním signálu se opakují tytéž hodnoty, a to nevede ke zlepšení popisu dynamiky systému.

Nyní se podíváme na odezvu na obdélníkový vstupní signál s proměnnou střídou a zároveň porovnáme s návrhem fuzzy systému, který byl navržen z odezvy nelineárního systému na tyto obdélníkové signály.



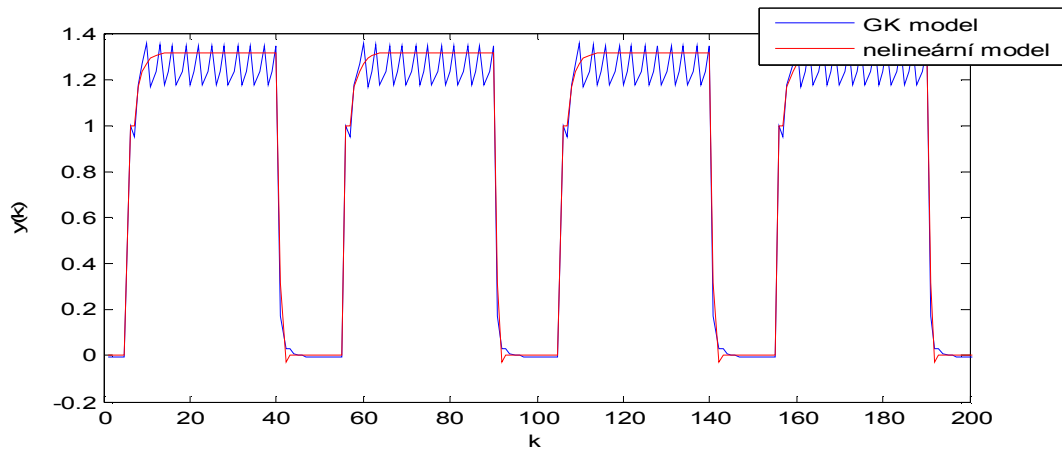
RMSE = 0.0361

Obr. 3.22 Odezva nelineárního modelu a fuzzy modelu - 10 pravidel (naučeného z 1000 náhodných dat) na obdélníkový vstup se střídou 20% , $T=50$ vzorků, $A=1$



RMSE = 0.0640

Obr. 3.23 Odezva nelineárního modelu a fuzzy modelu - 10 pravidel (naučeného z 1000 náhodných dat) na obdélníkový vstup se střídou 50% , $T=50$ vzorků, $A=1$



RMSE = 0.0787

Obr. 3.24 Odezva nelineárního modelu a fuzzy modelu - 10 pravidel (naučeného z 1000 náhodných dat) na obdélníkový vstup se střídou 70% , T=50 vzorků, A=1

V následující tabulce shrneme dosažené výsledky pro všechny navrhované modely.

vstupní obdélníkový signál	systém naučený z obdélníkového signálu			
	3 pravidla		7 pravidel	
	FCM	GK	FCM	GK
střída 20%	0.5328	0.1200	0.4057	0.1149
střída 50%	0.4279	0.2110	0.2851	0.2010
střída 70%	0.3421	0.2540	0.2512	0.2419

vstupní obdélníkový signál	systém naučený pro 200 dat - vstup náhodná posloupnost čísel					
	3 pravidla		7 pravidel		10 pravidel	
	FCM	GK	FCM	GK	FCM	GK
střída 20%	0.1166	0.1349	0.1245	0.1229	0.1473	0.2220
střída 50%	0.2045	0.1830	0.2121	0.2119	0.2220	0.2538
střída 70%	0.2462	0.2093	0.2544	0.2551	0.2600	0.2844

vstupní obdélníkový signál	systém naučený pro 1000 dat - vstup náhodná posloupnost čísel					
	3 pravidla		7 pravidel		10 pravidel	
	FCM	GK	FCM	GK	FCM	GK
střída 20%	0.1288	0.1879	0.1249	0.2903	0.1758	0.0361
střída 50%	0.1706	0.2280	0.1526	0.2280	0.2047	0.0640
střída 70%	0.1939	0.2516	0.1685	0.2138	0.2232	0.0787

Tab. 2 Chyba fuzzy modelů na obdélníkový vstupní signál T=50 vzorků, A=1

Kapitola 4

Závěr

V této práci jsme si ukázali návrh nelineárního systému pomocí dvou algoritmů shlukové analýzy. Jak dosažené výsledky naznačují, pro daný nelineární systém se nepodařilo najít fuzzy systém se stejnými vlastnostmi, ale ukázali jsme si, že lze dosáhnout přibližného popisu. Největší chybu do návrhu vnáší výsledná projekce shluků, kdy se dopouštíme určité chyby při samotné projekci hlavně v případě GK algoritmu, kde shluky neleží rovnoběžně se souřadnicovými osami, ale jsou různě pootočené.

Z hlediska návrhu záleží, pro jaký vstupní signál budeme systém používat a na základě toho se rozhodnout, jaký algoritmus použijeme. Pro náhodný signál a sinusový signál se jako nejlepší ukázal být návrh pomocí FCM algoritmu s 10 pravidly a pro obdélníkový signál návrh pomocí GK algoritmu pro 10 pravidel, v obou případech naučených z 1000 dat .

Na výsledný návrh také mají vliv použítá data pro učení systému, kde se ukázalo jako zcela nevhodné použít data pouze se dvěma stavy (obdélníkový signál), ale naopak použít signál s větší dynamikou. Pro zlepšení návrhu lze použít kombinaci vstupních signálů a zvolit dostatečný počet vzorků.

Výše zmíněné dva algoritmy tvoří základ algoritmů fuzzy shlukové analýzy. Ke zlepšení popisu modelu by mohlo vést vyzkoušení dalších algoritmů shlukové analýzy, případně modifikace stávajících algoritmů. Jako příklad uveďme Gath – Geva algoritmus, který k výpočtu vzdálenosti využívá exponenciální funkci, lineární a eliptické algoritmy, které využívají lineární závislost mezi daty. Poslední dva zmíněné algoritmy byly navrženy speciálně pro vícerozměrné shluky.

Literatura

- [1] JURA P., *Základy fuzzy logiky pro řízení a modelování*, Nakladatelství VUTIUM, Brno 2003
- [2] HANČ T., *Návrh Takagi – Sugeno fuzzy modelů*, Bakalářská práce, Fakulta elektrotechnická, ČVUT Praha 2005
- [3] *Data-driven fuzzy modeling*, http://fuzzy.iau.dtu.dk/tutor/ddfm_files/frame.htm, 2001
- [4] *Fuzzy clustering and data analysis toolbox*, <http://www.fmt.vein.hu/softcomp/>, 2002
- [5] HÖPPNER F., KLAWONN F., KRUSE R., RUNKLER T., *Fuzzy cluster analysis*, Nakladatelství John Wiley & Sons, 2000
- [6] ABONYI J., BABUŠKA R., SZEIFERT F., *Modified Gath-Geva Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models*, Hungary 2002