České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

Metody fuzzy shlukové analýzy při modelování nelineárních systémů

BAKALÁŘSKÁ PRÁCE

Vypracoval: Jan Procházka
Vedoucí práce: Ing. Petr Hušek, Ph.D.
Rok: 2008

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jan Procházka**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Metody fuzzy shlukové analýzy při modelování nelineárních systémů**

Pokyny pro vypracování:

1. Seznamte se se základními metodami fuzzy shlukové analýzy - fuzzy c-means algoritmus, Gustafson-Kesselův algoritmus.
2. Prověřte použitelnost těchto algoritmů při modelování nelineárních systémů zadaných vedoucím práce pomocí Takagi-Sugeno fuzzy modelů.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Petr Hušek, Ph.D.

Platnost zadání: do konce zimního semestru 2008/2009

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

L.S.

r.z. M. Demlová
doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 29. 2. 2008

Prohlášení

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

V Praze dne .....

….....

Jan Procházka

Poděkování

Na tomto místě bych chtěl poděkovat Ing. Petru Huškovi Ph.D. za poskytnutí podkladů k vypracování a za cenné rady. Také bych chtěl poděkovat rodině za podporu v době psaní této práce.

Jan Procházka

Abstrakt

Tato práce pojednává o metodách shlukové analýzy a její využití pro modelování nelineárních systémů. Konkrétně se zabývá postupy fuzzy c-means a Gustafson-Kesselovým algoritmem. Dále jsou zde rozebrány Takagi-Sugeno modely, jejich princip a sestavování pomocí výše uvedených fuzzy shlukových metod. Vytváření Takagi-Sugeno modelů je aplikováno na statickém nelineárním systému, kde se ověřuje schopnost aproximace a na dynamickém diskrétním nelineárním systému, kde algoritmy hledají fuzzy shluky ve čtyřech dimenzích.

Abstract

This paper explores fuzzy clustering methods and their usage for modeling nonlinear systems. We discuss methods fuzzy c-means and Gustafson-Kessel algorithm. Further we focused Takagi-Sugeno models, their principle and assembling using fuzzy clustering methods quoted above. Takagi-Sugeno models were tested on static nonlinear system where we verified aproximation capability and on dynamic discrete nonlinear system, where algorithms were searching for fuzyy clusters in four dimensions.

Obsah

1 Úvod **7**

2 Fuzzy shluková analýza **9**

2.1 Fuzzy c-means....11

2.2 Gustafson-Kesselův algoritmus....13

2.3 Určení počtu shluků....15

2.4 Další algoritmy fuzzy shlukové analýzy....15

3 Modelování nelineárních systémů **17**

3.1 Nelineární systémy....17

3.2 Takagi-Sugeno pravidla....17

3.3 Princip inference....19

3.4 Vytváření pravidel pro TS modely....21

4 Aplikace algoritmů **24**

4.1 Porovnání kvality modelů....24

4.2 Analýza statického systému....25

4.3 Analýza dynamického systému....30

5 Závěr **36**

Reference **37**

Obsah CD **38**

1 Úvod

Člověk, jako každý jiný živý tvor, má potřebu zkoumat svět kolem sebe. Už od velmi raného věku jako miminko leze všude po pokoji, osahává hračky, všechen nábytek, který mu přijde do cesty a který ještě nezná. Nejspíš je to dáno evolucí, kdy se každý živočich musel seznámit s okolím, aby v něm mohl žít a přežít. Také už před stovkami let lidé začali nejenom zkoumat, ale popisovat své okolí. Ať už v jakékoliv podobě. Spisovatelé popisovali příběhy slovy běžného jazyka, herci k tomu přidali výjevy, gestikulaci a mimiku a matematici se snažili popsat svět pomocí pojmů a rovnic, které dávaly nějaký smysl.

Ze začátku šlo o jednoduché představy, které se postupem času čím dál tím více zpřesňovaly. Příkladem může být představa o planetě Zemi. Ze začátku lidé věřili, že Země je placka, poté si všimli, že žijí na kouli, když následně zjistili, že se nejedná o pravidelnou kouli, ale o kouli na pólech zploštělou. Přesnějším a přesnějším měřením bychom mohli zjistit, jak moc je Země zploštělá, nejspíš i v relativně malých jednotkách. Na co ovšem potřebujeme znát zploštění Země v metrech? Nebo dokonce v centimetrech? Nebo v ještě menších jednotkách? Ve většině případů stačí říct, že Země je koule *trochu* zploštělá. Každý si představí jak je asi zploštělá, i když nemá přesný údaj. Takto jednoduše lze popisovat všechny věci kolem nás. Každý z nás používá přirozený jazyk, který taková slova jako *malý*, *trochu*, *někdy*, *častý*, *velký* obsahuje. Těmto slovům se říká vágní pojmy, protože si i pod stejným pojmem můžeme představovat různá určení velikosti, četnosti apod.

Fyzika se snaží popisovat věci co nejpřesněji a vyvozovat přesné závěry. Jenomže nikdy nelze fyzikální veličinu změřit zcela přesně. Pokud bychom se dostali do velmi malých měřítek, tak stejně s přesným popisem 100% neobstojíme, protože jednak by byl popis velmi složitý a špatně pochopitelný a jednak do hry vstupuje neurčitost v podobě kvantových jevů.

Lidé užívají vágní pojmy k popisu věcí a událostí. Většině žáků v autoškole stačí říct, když se blíží ke křižovatce, kde chceme odbočovat, aby *více* přibrzdili. Není potřeba jim přesně sdělit o jakou vzdálenost mají sešlápnout pedál brzdu. Spíše by to uškodilo dopravní situaci nebo autu školitele.

Nabízí se tedy možnost popisovat okolí, kde je všudypřítomná neurčitost, právě vágními pojmy i v případě, že chceme nějaký systém řídit anebo pouze předpovídat jeho chování. V 70. letech přišel L.A Zadech v [1] s fuzzy logikou, která právě tyto

„rozmlžené“ pojmy postihuje. Následně vzniklo několik teorií o modelování systémů, o jejich řízení a předpovídání právě s využitím fuzzy logiky.

Jednou z teorií jak modelovat systémy, jsou Takagi-Sugeno modely, které rozebírá tato práce. Dají se použít v případě, že systém je popsán expertem v kombinaci přirozeného jazyka s jednoduchými lokálními funkcemi anebo častěji, když o fungování systému víme málo informací a můžeme sledovat pouze jeho vstupy a výstupy. Pak se o vytváření Takagi-Sugeno modelů stará fuzzy shluková analýza, která je náplní další kapitoly.

2 Fuzzy shluková analýza

Pokud by nám byla předložena jakákoliv dvoudimenzionální data a měli bychom za úkol najít shluky, člověk se „koukne a vidí“, když není rozložení složité. Problém ale nastane v okamžiku, kdy je zapotřebí znát velmi rychle skupiny podobných dat anebo jsou vektory vstupních údajů vícedimenzionální.

Tyto problémy lze řešit nebo alespoň zmenšit pomocí shlukové analýzy. Klasická shluková analýza rozdělí data do skupin. Vstupní vektory jsou přiřazeny do shluků jednoznačně. Buď vstupní údaj do shluku patří nebo nepatří. Tím se ovšem ztrácí část informace. Je jasné, že vstupní vektor, který leží uprostřed shluku bude nejspíš důležitější (více patřit shluku), než vstupní vektor, který leží na okraji. Problém s překrýváním fuzzy shluková analýza řeší. Dovoluje libovolnému vstupnímu vektoru, aby za určitých podmínek, které jsou rozebrány níže, patřil do různých shluků, ke každému s určitým stupněm příslušnosti.

Existují dva přístupy k fuzzy shlukové analýze - probabilistický a posibilistický. Oba přístupy jsou podobné, přiřazují každému vstupnímu údaji stupeň příslušnosti v rozmezí $\langle 0,1 \rangle$. Stupeň příslušnosti 0 říká, že údaj určitě neleží v daném shluku, naopak 1 jednoznačně určuje přiřazení k dané skupině dat. My budeme používat probabilistický přístup, který pro nezašumělá data vrací většinou lepší výsledky. Posibilistický přístup oproti probabilistickému není limitován podmínkou, že součet příslušností určitého údaje ke všem shlukům musí být roven jedné. Z čehož vyplývá například to, že interpretace rozdělení dat do shluků se liší. Probabilistické rozdělení říká, s jakou pravděpodobností patří vstupní údaj do určitého shluku. Oproti tomu posibilistické rozdělení určuje jak moc, na stupnici od nuly k jedné, daný údaj patří určitému shluku. Více podrobností a porovnání obou přístupů lze najít v literatuře [2].

Máme tedy N vstupních dat z množiny $\boldsymbol{Z} = [z_1, \ldots. z_N]$ zapsané maticově a chceme ji rozdělit do c shluků. Každý vstupní vektor může být n-dimenzionální a bude mít následující tvar $\boldsymbol{z_k} = (z_{k,1}, \ldots, z_{k,n})^T$. K tomu, abychom mohli určit, do jaké skupiny data patří, musíme je umět od sebe rozlišit. Pro to, zda jsou si dva údaje v nějaké míře podobné, potřebujeme nástroj, který data porovná a podobnost vyhodnotí. Slouží k tomu funkce vzdálenosti. Nejčastěji se používá Euklidova nebo manhattanská vzdálenost, obecně ale můžeme použít jakoukoliv funkci $d(z_i, z_j)$, která splňuje následující tři pravidla:

$$d(z, z) = 0 \tag{2.1}$$

$$d(z_1, z_2) = d(z_2, z_1) \tag{2.2}$$

$$d(z_1, z_3) \leq d(z_1, z_2) + d(z_2, z_3) \tag{2.3}$$

Dále pomocí fuzzy shlukové analýzy lze každému údaji z_k ze $\boldsymbol{Z}$ přiřadit míru příslušnosti, nakolik do každé ze skupin patří. Přiřazení lze provést maticí příslušnosti $\boldsymbol{U}$, jejíž počet řádků je dán počtem shluků c, počet sloupců počtem vstupních dat N a čísla matice $u_{i,k} \in \langle 0,1 \rangle$ udávají míru příslušnosti z_k do shluku $\boldsymbol{V}_i$. Matice příslušnosti $\boldsymbol{U}$ musí splňovat následující podmínky:

$$\sum_{i=1}^{c} u_{i,k} = 1 \; ; \quad k = 1, \ldots, N \tag{2.4}$$

$$0 < \sum_{k=1}^{N} u_{i,k} < N \; ; \quad i = 1, \ldots, c \tag{2.5}$$

Podmínka (2.4) říká, že vstupní vektor z_k musí patřit nejméně do jednoho shluku a že součet příslušností každého údaje ke všem shlukům musí být roven jedné (probabilistický přístup). Nerovnost (2.5) přiřazuje každému shluku alespoň jeden vstupní údaj a zároveň říká, že není možné, aby byla všechna vstupní data přiřazena do jedné skupiny s příslušností 1 (tedy jednoznačně).

Fuzzy shlukové algoritmy založené na fuzzy c-means se snaží minimalizovat funkcionál (objektivní funkci),

$$J_m(\boldsymbol{U}_c) = \sum_{i=1}^{c} \sum_{k=1}^{N} \left(u_{i,k}\right)^m d^2(z_k, \boldsymbol{v}_i) \tag{2.6}$$

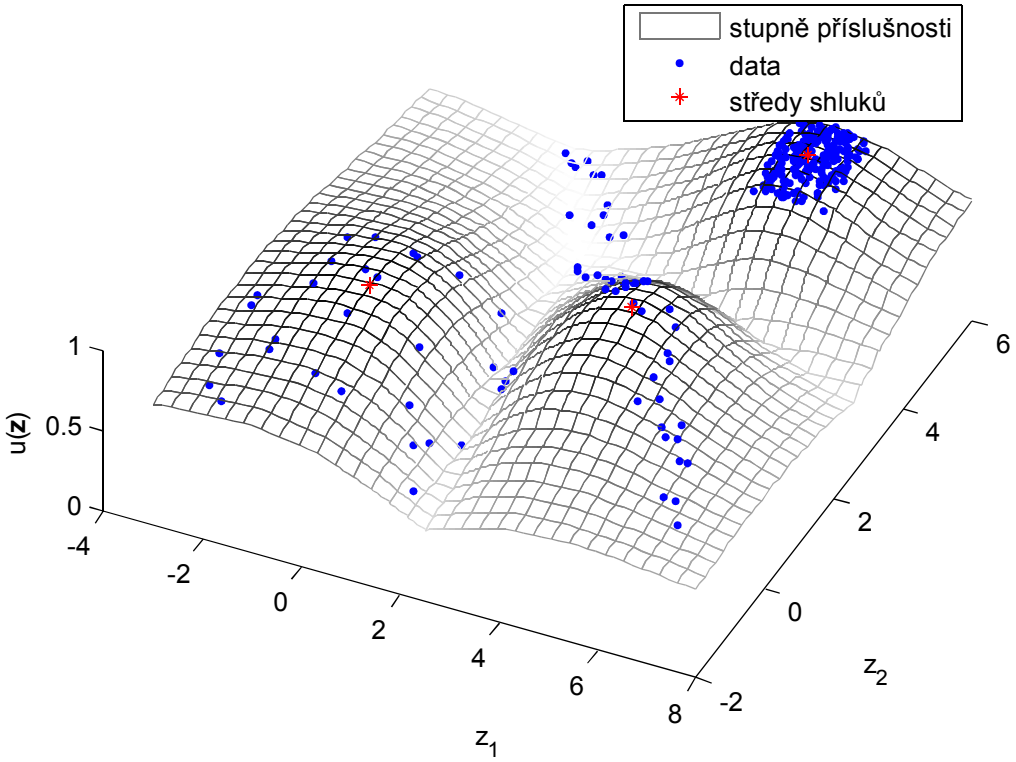
kde $\boldsymbol{v}_i$ je střed shluku $\boldsymbol{V}_i$ a m je fuzzyfikační konstanta. Jedná se vlastně o minimalizaci odchylky u metody vážených nejmenších čtverců vzdáleností dat od středu shluku. Fuzzyfikační konstanta m určuje, jak moc budou shluky rozostřené. Pro parametr m blížící se shora k 1 přechází fuzzy shluková analýza na klasickou shlukovou analýzu s ostrými shluky a pro m blížící se ∞ budou mít všechna data stejný stupeň příslušnosti ke všem shlukům. Většinou se nastavuje m = 2. Pokud je tato objektivní funkce minimalizována, je dosaženo optimálního řešení. Pokud máme najít minimum objektivní funkce J_m, je nutné aby její první parciální derivace byly 0 (lokální minimum) a výsledkem řešení je tedy

$$u_{i,k} = \frac{1}{\sum_{j=1}^{c} \left[\frac{d(\boldsymbol{z}_k, \boldsymbol{v}_i)}{d(\boldsymbol{z}_k, \boldsymbol{v}_j)} \right]^{\frac{2}{m-1}}} \tag{2.7}$$

Existuje mnoho dalších strategií, jak dosáhnout optimálního rozdělení dat. Další příklad minimalizace objektivní funkce je shlukování s šumem. Do sady shluků je přidána další speciální skupina, kam se řadí údaje, které jsou od původních shluků příliš vzdálené a jsou považovány za šum. Lze najít i další, zcela odlišné strategie rozdělení dat do shluků, jako jsou například genetické algoritmy nebo neuronové sítě.

2.1 Fuzzy c-means

Fuzzy c-means je základní metodou z rodiny fuzzy shlukových algoritmů, která rozpoznává shluky kruhových tvarů. Pro tento typ algoritmů je nutné znát předem počet shluků $c > 2$, do kterých se budou data zařazovat. Problém určení počtu shluků se diskutuje v části 2.3. Názornou ukázku fuzzy c-means algoritmu lze najít na obrázku 2.1, kde vidíme, jak algoritmus určil středy shluků a rozložení stupňů příslušností pro určenou oblast. Kvůli tomu, že fuzzy c-means dokáže rozpoznávat pouze kruhové shluky, tato metoda si neporadila dobře s prostředním shlukem ve tvaru přímky.



obr 2.1: Detekce shluků s fuzzy c-means

Fuzzy c-means je iterativní algoritmus a probíhá v následujících krocích:

1. Nejdříve je nutné zvolit parametr m tak, že $m > 1$ a počáteční rozklad množiny dat (počáteční nastavení $\boldsymbol{U}^{(0)}$), který může být při respektování podmínek (2.4) a (2.5) náhodný. Dále požadovanou přesnost $\varepsilon > 0$, kdy algoritmus končí, pokud je jí dosaženo. Nakonec se nastaví počítadlo kroků $l = 0$.
2. Pro každý shluk $\boldsymbol{V}_i^{(l)}$ vypočítáme nové centrum podle vzorce

$$\boldsymbol{v}_i^{(l)} = \frac{\sum_{k=1}^{N} \left[u_{i,k}^{(l)}\right]^m \boldsymbol{z}_i}{\sum_{k=1}^{N} \left[u_{i,k}^{(l)}\right]^m} \; ; \quad i = 1, \ldots, c \tag{2.8}$$

   kde $\boldsymbol{v}_i^{(l)}$ je vektor středu shluku v kroku l, který je počítán *po složkách*.
3. Upravíme matici příslušnosti $\boldsymbol{U}^{(l+1)}$ pro každé $k = 1, \ldots, N$ tak, že
   (a) pro všechna $i = 1, \ldots, c$, pro která platí, že $d(\boldsymbol{z}_k, \boldsymbol{v}_i) > 0$ nastavíme

$$u_{i,k}^{(l+1)} = \frac{1}{\sum_{j=1}^{c} \left[\frac{d(\boldsymbol{z}_k, \boldsymbol{v}_i^{(l)})}{d(\boldsymbol{z}_k, \boldsymbol{v}_j^{(l)})}\right]^{\frac{2}{m-1}}} \tag{2.9}$$

   (b) pro všechna $i = 1, \ldots, c$, pro která platí, že $d(\boldsymbol{z}_k, \boldsymbol{v}_k) = 0$ položíme $u_{i,k}^{(l+1)}$ rovna libovolným nezáporným číslům tak, aby byla splněna podmínka (2.4) a pro ostatní i položíme $u_{i,k}^{(l+1)} = 0$
4. Porovnáme výsledek současného rozkladu $l + 1$ oproti předchozímu z kroku l. To lze provést více způsoby, níže je uveden jeden z možných.

$$e = max_{k=1,\ldots,N;\, i=1,\ldots,c} \left|u_{i,k}^{(l+1)} - u_{i,k}^{(l)}\right| \tag{2.10}$$

   Pokud je $e < \varepsilon$ algoritmus skončí, jinak zvýšíme počítadlo kroků l o 1 a pokračujeme opět krokem 2.

Výsledkem fuzzy c-means algoritmu je tedy matice $\boldsymbol{U}^{(l)}$, která obsahuje příslušnosti jednotlivých vektorů do jednotlivých shluků $\boldsymbol{V}_i^{(l)}$ se středy $\boldsymbol{v}_i^{(l)}$.

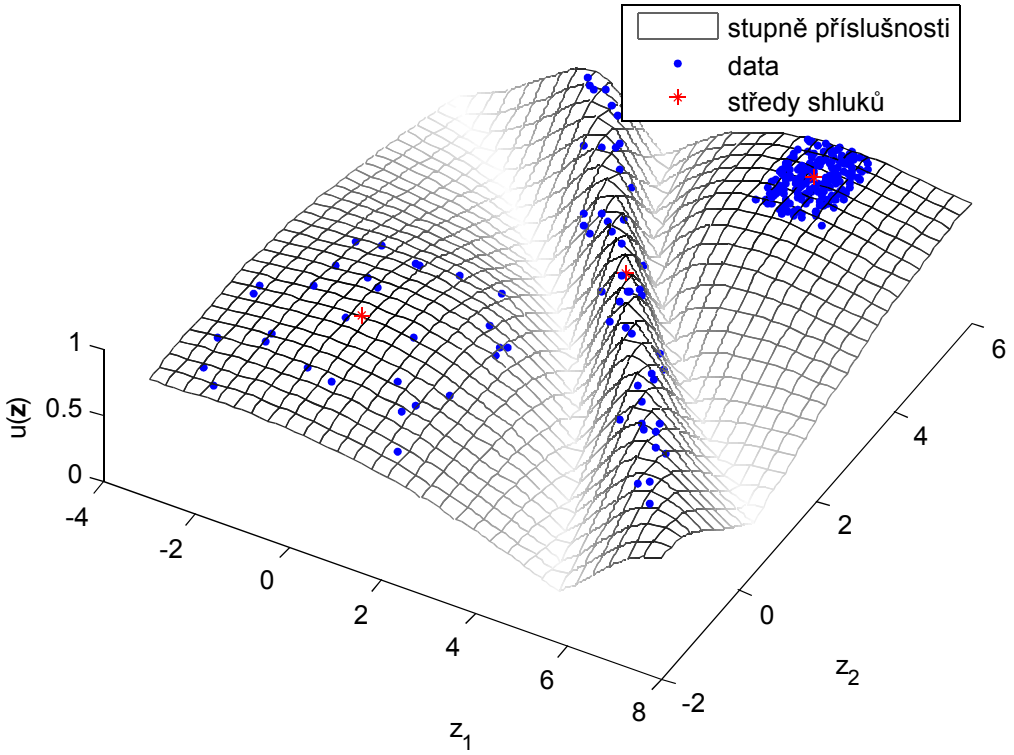
2.2 Gustafson-Kesselův algoritmus

Gustafson-Kesselův algoritmus je upravený fuzzy c-means tak, aby byl schopen rozpoznat shluky různých tvarů a orientací. Shluky v této metodě jsou oproti fuzzy c-means elipsoidní a navíc reprezentovány symetrickou pozitivně definitní maticí $\boldsymbol{P}_i$. Někdy se této matici říká deformační matice, protože upravuje tvar shluků, který mají ve dvourozměrném prostoru tvar elipsy. K výpočtu $\boldsymbol{P}_i$ se používá kovarianční matice $\boldsymbol{S}_i$.

$$\boldsymbol{S}_i = \frac{\sum_{k=1}^{N} \left(u_{i,k}\right)^m (\boldsymbol{z}_k - \boldsymbol{v}_i)^T (\boldsymbol{z}_k - \boldsymbol{v}_i)}{\sum_{k=1}^{N} \left(u_{i,k}\right)^m} \tag{2.11}$$

$$\boldsymbol{P}_i = [det(\boldsymbol{S}_i)]^{\frac{1}{N}} \boldsymbol{S}^{-1} \tag{2.12}$$

Dále je pro $\boldsymbol{P}_i$ definováno omezení $det(\boldsymbol{P}_i) = \rho$, kde ρ je konstanta daná pro každou matici. Tímto je určen objem shluků. Natočení jejich poloos je určeno vlastními vektory matice $\boldsymbol{P}_i$ a velikosti jejich poloos odpovídají odmocninám z vlastních čísel matice $\boldsymbol{P}_i$.



obr 2.2: Detekce shluků Gustafson-Kesselovým algoritmem

Na obrázku 2.2 je znázorněna detekce shluků Gustafson-Kesselovým algoritmem. Jedná se o stejná data, jako na ukázce 2.1 v předchozí sekci. Gustafson-Kesselův algoritmus si poradil velmi dobře s určením shluků. I prostřední shluk, který dělal fuzzy c-means problémy, byl nalezen správně (tak jak bychom ho intuitivně určili my).

V následujících krocích je naznačen postup rozdělení vstupních dat do shluků:

1. Na začátku nastavíme parametry a hodnoty stejně jaku u fuzzy c-means algoritmu. Zvolíme si tedy $m > 1$, $\varepsilon > 0$ a nastavíme $\boldsymbol{U}^{(0)}$ a $l = 0$.
2. Pro každý shluk $\boldsymbol{V}_i$ vypočítáme střed $\boldsymbol{v}^{(l)}$ podle (2.8).
3. Dále spočteme pro všechny shluky kovarianční matice $\mathbf{S}_i^{(l)}$ spolu s deformačními maticemi $\boldsymbol{P}_i^{(l)}$ podle rovnic (2.11) a (2.12).
4. Pro každé $k = 1, \ldots, N$ nastavíme vzdálenosti

$$d_P(z_k, \boldsymbol{v}_k^{(l)}) = \sqrt{(z_k - \boldsymbol{v}_i^{(l)})\boldsymbol{P}_i^{(l)}(z_k - \boldsymbol{v}_i^{(l)})^T}\ ; \quad i = 1, \ldots, c \tag{2.13}$$

5. Pro každé $k = 1, \ldots, n$ upravíme matici příslušnosti $\boldsymbol{U}^{(l+1)}$ obdobně jako u fuzzy c-means s tím rozdílem, že jako funkci vzdálenosti použijeme d_P tedy:
   (a) pro všechna $i = 1, \ldots, c,$ pro která platí, že $d_P(z_k, \boldsymbol{v}_i) > 0$ nastavíme

$$u_{i,k}^{(l+1)} = \frac{1}{\sum_{j=1}^{c}\left[\frac{d_P(z_k, \boldsymbol{v}_i^{(l)})}{d_P(z_k, \boldsymbol{v}_j^{(l)})}\right]^{\frac{2}{m-1}}} \tag{2.14}$$

   (b) pro všechna $i = 1, \ldots, c$, pro která platí, že $d_P(z_k, \boldsymbol{v}_i) = 0$ položíme $u_{i,k}^{(l+1)}$ rovna libovolným nezáporným číslům tak, aby byla splněna podmínka (2.4) a pro ostatní i položíme $u_{i,k}^{(l+1)} = 0$
6. Porovnáme rozklady například podle (2.10), a když $e < \varepsilon$ algoritmus skončí, jinak se zvýší počítadlo l o 1 a přejdeme na krok 2.

Výsledkem Gustafson-Kesselova algoritmu je matice příslušnosti $\boldsymbol{U}^{(l)}$, ve které jsou obsaženy stupně příslušnosti vstupních dat do nalezených shluků $\boldsymbol{V}_i^{(l)}$ se středy $\boldsymbol{v}_i^{(l)}$ a deformačními maticemi $\boldsymbol{P}_i^{(l)}$.

2.3 Určení počtu shluků

Jak již bylo v předchozích sekcích naznačeno, fuzzy c-means a Gustafson-Kesselův algoritmus musí předem znát počet shluků, do kterých budou data zařazována. Často se stane, že počet skupin neznáme a musíme tento problém řešit.

Jednou z možností je definovat si funkci, která ohodnotí kvalitu rozložení dat do shluků pro celý rozklad. Dále si stanovíme horní hranici počtu skupin $c_{\max}$. Za použití fuzzy shlukových algoritmů rozdělíme data do tříd pro počty $c = 2, \dots, c_{\max}$ a pro každé takové rozdělení ohodnotíme definovanou funkcí kvality. Jedna z možných charakteristik je následující.

$$PC(\boldsymbol{U}_c) = \frac{\sum_{i=1}^{c} \sum_{k=1}^{N} (u_{i,k})^2}{N} \tag{2.15}$$

Tato funkce nabývá hodnot $\frac{1}{c} \leq PC(\boldsymbol{U}_c) \leq c$ a vyšší hodnoty znamenají lepší rozdělení dat do shluků. Existuje mnoho dalších funkcí kvality, které se buď maximalizují nebo minimalizují. Pokud můžeme, je dobré využít více charakteristik kvality a vhodně je kombinovat.

Další možností je posuzovat jednotlivé shluky odděleně podle funkcí kvality shluku. Opět si stanovíme horní hranici počtu shluků $c_{\max}$ a provedeme fuzzy shlukový algoritmus pro tento počet shluků. Výsledné skupiny porovnáme, podobné spojíme a velmi špatné odstraníme. Tento postup opakujeme, dokud neexistují žádné podobné ani žádné špatné shluky.

Detailnější rozbor problému určení počtu shluků je možné prostudovat v literatuře [2], kde je uvedeno a porovnáno mnoho globálních i lokálních funkcí kvality.

2.4 Další algoritmy fuzzy shlukové analýzy

V kapitolách 2.1 a 2.2 jsme popsali dva základní algoritmy fuzzy shlukové analýzy. Na podobném principu funguje ještě další algoritmus Gath-Geva, který rozšiřuje Gustafson-Kesselův postup. Gath-Geva algoritmus bere v úvahu velikost i hustotu shluků.

Další algoritmy jako fuzzy c-spherical/elipsoidal/quadratic shells umožňují detekci

určitých obrazců – elips, obdélníků, polygonů a jiných. Přičemž existují i adaptivní verze těchto postupů, které se umí přizpůsobit vstupním datům. Opět si čtenář může prostudovat tyto algoritmy v literatuře [2].

3 Modelování nelineárních systémů

3.1 Nelineární systémy

V praxi se běžně kolem nás vyskytují nelineární systémy. Mnoho z nich chceme řídit nebo alespoň předpovídat jejich chování v budoucnosti. Příkladem může být, pro každého z nás zcela běžná věc, předpověď počasí. Jako u většiny nelineárních systémů se zde odráží několik věcí. Nejsme schopni zcela přesně změřit relevantní veličiny i v případě, že by nebyly zatíženy žádným šumem. Často ale pouze malá změna vstupní veličiny způsobí rozsáhlé přeměny v budoucích stavech systému. Dále obecně nelze sestavit přesný model nelineárního systému. Pokud bychom mohli sestavit přesný model, většinou bude hodně složitý a neefektivní. Takže se ho budeme snažit různými způsoby zjednodušit, i za cenu toho, že dále do procesu zanášíme chyby.

Nelineární systém lze popsat soustavou nelineárních diferenciálních rovnic, u kterých obecně nelze najít analytické řešení. Neplatí zde princip superpozice jako u lineárních systémů, takže většinou nemůžeme problém rozdělit přímo na menší podproblémy, opět, bez ztráty na přesnosti někdy až použitelnosti.

S klasickými modely si tedy mnohdy nevystačíme. L. A. Zadeh v [1] položil základy fuzzy logiky, která umožňuje zachycovat nepřesnosti měření a vágnost pojmů. Díky ní lze vytvářet fuzzy modely, které mohou dostatečně přesně a robustně popisovat reálné systémy.

Jedny z nejběžnějších a nejsnáze použitelnějších prostředků pro fuzzy modelování jsou Takagi-Sugeno modely, kterými se zabývá následující část.

3.2 Takagi-Sugeno pravidla

V této sekci se zaměříme na popis Takagi-Sugeno pravidel, která definují Takagi-Sugeno model. Pokud bude v textu zmínka o soustavě Takagi-Sugeno pravidel a o fuzzy modelu, jedná se v kontextu této práce o pojmy stejného významu. Obecně fuzzy model může být i jiného typu, například Mamdaniho modely a mnoho dalších, včetně velmi zajímavých neuro-fuzzy modelů testovaných v [6].

Budeme modelovat neznámý diskrétní nelineární systém,

$$y = f(\boldsymbol{x}) \tag{3.1}$$

kde $\boldsymbol{x} = (x_1, x_2, \ldots x_n)$ vstupní vektor a y je výstup systému. Číslo n tedy udává počet vstupů. Pokud je nelineární systém dynamický, některá $\boldsymbol{x}_k$ z $\boldsymbol{x}$ obsahují výstupy z předchozích měření jako zpětnou vazbu. Kdybychom chtěli modelovat spojitý dynamický nelineární systém, bylo by nutné pečlivě zvolit vzorkovací frekvenci a počty vstupů ze zpětné vazby. Tento úkol je obecně obtížný a tato práce se jím nezabývá. Modelem systému budeme myslet

$$y^A = f^A(\boldsymbol{x}) \tag{3.2}$$

Takagi-Sugeno modely jsou charakterizovány soustavou pravidel jestliže-pak nebo také jinými slovy soustavou lokálních modelů. Tvar pravidla je uveden níže

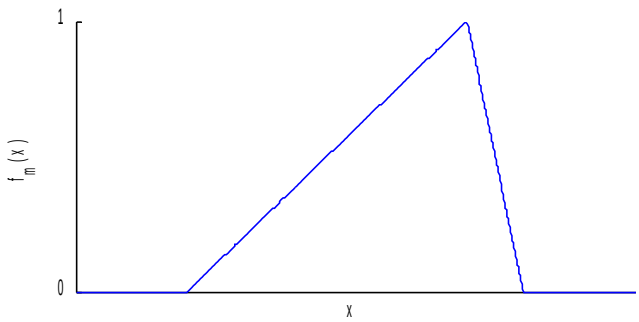
$$R_i: \ \textbf{Jestliže } \boldsymbol{x} \text{ je } \boldsymbol{A}_i(\boldsymbol{x}) \ \textbf{Pak } y_i^A = f_i^A(\boldsymbol{x}) \ ; \ [w_i] \tag{3.3}$$

kde $\boldsymbol{x}$ je vstupní vektor, y_i^A reprezentuje výstupní veličinu lokálního modelu, $\boldsymbol{A}_i(\boldsymbol{x})$ vektor fuzzy množin, které jsou reprezentované funkcemi příslušnosti. Navíc se pravidlu přiřazuje důvěryhodnost w_i, která udává, jak moc v pravidlo věříme. Důvěryhodnost pravidla určuje expert. V případě, že není možné váhu zjistit, nastavuje se $w_i = 1$ pro každé pravidlo. Pravidlo (3.3) lze pro jednotlivé složky rozepsat na následující tvar

$$R_i: \ \textbf{Jestliže } x_1 \text{ je } A_{i,1}(x_1) \text{ a } \ldots \text{ a } x_n \text{ je } A_{i,n}(x_n) \ \textbf{Pak } y_i^A = f_i^A(\boldsymbol{x}) \ ; \ [w_i]$$

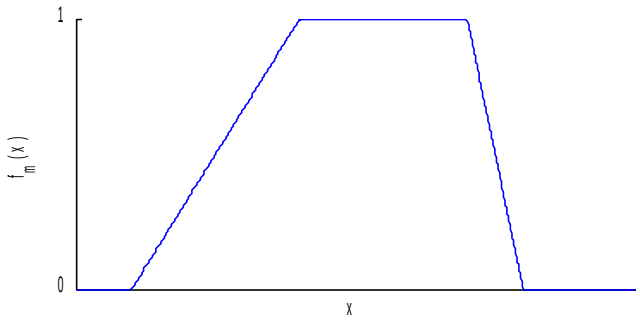
ve kterém je vidět, že předpoklady jsou spojovány sjednocením. Je zde také patrné, že každá složka vstupního vektoru se může ohodnocovat různou funkcí příslušnosti. Dále je uvedeno několik nejběžnějších typů těchto funkcí, kterými je reprezentována fuzzy množina $A_{i,j}(x_j)$.

- Trojúhelníková funkce příslušnosti



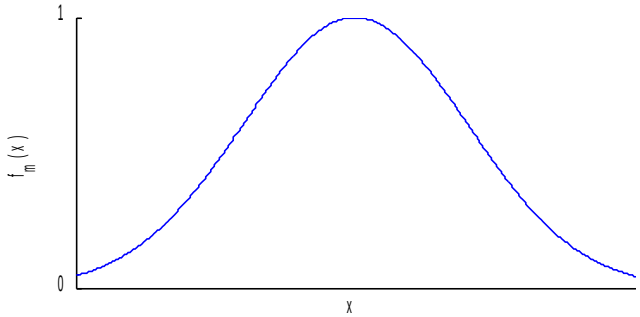
obr 3.1: Trojúhelníková funkce příslušnosti

- Lichoběžníková funkce příslušnosti



obr 3.2: Lichoběžníková funkce příslušnosti

- Funkce příslušnosti reprezentovaná gaussovou křivkou



obr 3.3: Funkce příslušnosti reprezentovaná gaussovou křivkou

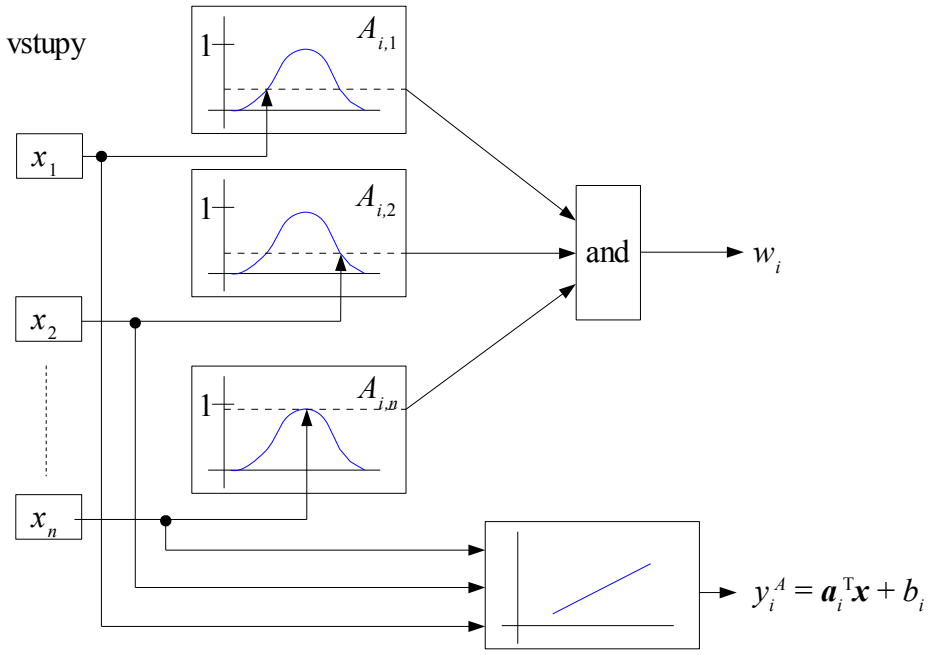
Funkce $f_i^A(\boldsymbol{x})$ v závěru pravidla (3.3) může být obecně jakákoliv funkce pro $\boldsymbol{x}$. Výhodný a velmi často používaný tvar této funkce je lineární kombinace vstupního vektoru s posunutím. Dále v práci budeme předpokládat a používat právě tento tvar.

$$f_i^A(\boldsymbol{x}) = \boldsymbol{a}_i^T \boldsymbol{x} + b_i \tag{3.4}$$

V následující části se budeme věnovat principu inference, tedy jak Takagi-Sugeno pravidla modelují nelineární systém.

3.3 Princip inference

Princip inference říká, jak jsou vstupní data přepočítávána pomocí Takagi-Sugeno pravidel na výstup. Máme vstupní vektor $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$, soustavu pravidel a chceme odhadnout výstup systému pomocí již sestaveného modelu.



obr 3.4: Postup výpočtu výstupu z lokálního modelu pomocí Takagi-Sugeno pravidla

1. Pro každé pravidlo $i = 1, \ldots, c$ se spočte výstup a jeho váha. Metodu výpočtu nastiňuje obr. 3.4 a postupuje se podle podle následujících kroků:

   (a) Nejdříve se jednotlivé vstupní složky x_j z $\boldsymbol{x}$, které jsou obsaženy v pravidle, fuzzyfikují (tzn. přiřadí se jim hodnota funkce příslušnosti $u_{x_j} = A_{i,j}(x_j)$).

   (b) Dále se provede průnik příslušností u_{x_k} pro všechna $j = 1, \ldots, n$, zváží se mírou důvěryhodnosti w_i a dostaneme míru naplnění pravidla $\beta_i(\boldsymbol{x})$.

$$\beta_i(\boldsymbol{x}) = w_i \boldsymbol{A}_i(\boldsymbol{x}) = w_i \prod_{j=1}^{n} A_{i,j}(x_j) \tag{3.5}$$

   (c) Následně se spočítá výstup y_i^A z pravidla podle

$$y_i^A = \boldsymbol{a}_i^T \boldsymbol{x} + b_i \tag{3.6}$$

2. Nakonec se z (3.5) a (3.6) sestaví výstup z globálního modelu (ze všech pravidel) pomocí přibližné dedukce, která je vhodná při aproximaci funkcí.

$$f^A(\boldsymbol{x}) = \frac{\sum_{i=1}^{c} \beta_i(\boldsymbol{x})\, y_i^A}{\sum_{i=1}^{c} \beta_i(\boldsymbol{x})} \tag{3.7}$$

3.4 Vytváření pravidel pro TS modely

Takagi-Sugeno modely, potažmo Takagi-Sugeno pravidla lze sestavit dvěma způsoby. Můžeme využít znalostí experta, který popíše chování systému v lokálních oblastech. Příkladem poslouží opět předpověď počasí. Expert může navrhnout jednoduché pravidlo

$$R: \text{ Jestliže } (\Delta\, p_a) \text{ je } (\textit{kladná vysoká}) \text{ Pak } (t_{+1}) = a\,\Delta\, p_a + b\ ; \quad w = 1$$

kde p_a je atmosférický tlak jako vstupní a t_{+1} je teplota vzduchu za 1 hodinu jako výstupní veličina. Konstanty a a b jsou rovněž určeny expertem, který ovšem musí znát charakteristiku systému ve zkoumaném bodě. Soustavou podobných lokálních modelů je možné sestavit model globální systému. Většinou se ovšem při vytváření Takagi-Sugeno pravidel využívá jiné metody popsané v následujícím odstavci.

Další možností k sestavování pravidel je využití analýzy dat. Nejdříve získáme informace o systému pozorováním vstupních veličin z $\boldsymbol{X}$ a odpovídajícím výstupům z $\boldsymbol{y}$.

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_N^T \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad \boldsymbol{Z} = [\boldsymbol{X} \quad \boldsymbol{y}] \tag{3.8}$$

Data $\boldsymbol{Z}$ roztřídíme do skupin. K rozdělení do tříd lze využít fuzzy shlukové analýzy, jejíž dvě metody jsou popsány v kapitole 2. Ze shluků pak můžeme vytvořit soustavu pravidel (3.3).

Vytváření pravidel by se dalo přirovnat k učení neuronových sítí bez učitele. Zde také ze vstupních dat formujeme konstanty a tvary funkcí. Významný rozdíl je mimo jiné v tom, že Takagi-Sugeno pravidla jsou snáze interpretovatelná.

Soustava Takagi-Sugeno pravidel (3.3) bude následující,

$$\begin{aligned} R_1: &\ \text{Jesliže } \boldsymbol{x} \text{ je } \boldsymbol{A}_1(\boldsymbol{x}) \text{ Pak } y_1^A = \boldsymbol{a}_1^T \boldsymbol{x} + b_1 \\ &\ \vdots \\ R_c: &\ \text{Jesliže } \boldsymbol{x} \text{ je } \boldsymbol{A}_c(\boldsymbol{x}) \text{ Pak } y_c^A = \boldsymbol{a}_c^T \boldsymbol{x} + b_c \end{aligned} \tag{3.9}$$

kde c je počet shluků, které jsme získali z fuzzy shlukové analýzy.

Jinými slovy poté, co se data zpracují metodou fuzzy shlukové analýzy pro $\boldsymbol{Z}$, získáme počet shluků c, matici příslušnosti $\boldsymbol{U}$, shluky $\boldsymbol{V}_i$ s odpovídajícími středy $\boldsymbol{v}_i$, popřípadě deformační matici $\boldsymbol{P}_i$ nebo jinou charakteristiku shluku závisející na zvolené metodě pro $i = 1, \ldots, c$. Pro naše účely postačí vektory $\boldsymbol{v}_i^x$, což jsou středy shluků $\boldsymbol{v}_i$ bez poslední

souřadnice (viz rozměr $\boldsymbol{Z}$ v (3.8)) Počet Takagi-Sugeno pravidel pak odpovídá počtu nalezených shluků. Budeme tedy hledat množinu funkcí příslušností $\boldsymbol{A}_i(\boldsymbol{x})$ a koeficienty $\boldsymbol{a}_i^{\mathrm{T}} = (a_{i,1}, \ldots, a_{i,n})$ a b_i.

Funkce příslušnosti $\boldsymbol{A}_i(\boldsymbol{x})$ lze najít pomocí projekce funkcí příslušnosti na shluky $\boldsymbol{V}_i$ pro každou souřadnici zvlášť. Maximum funkce příslušnosti bude odpovídat středu shluku a směrem od maxima budou funkční hodnoty klesat. Klesání je dáno tím, jaký tvar funkce příslušnosti jsme k projekci zvolili a jaký tvar a velikost má odpovídající shluk.

Shluky fuzzy c-means algoritmu jsou kruhové, ale nemají určenou velikost. Tento problém lze obejít. Například pokud budeme využívat Gaussovy funkce příslušnosti, je možné její rozptyl spočítat pomocí střední kvadratické odchylky vzdálenosti dat od středu shluku.

$$\sigma_{i,j}^2 = \frac{\sum_{k=1}^{N} u_{i,k}(x_{k,j} - v_{i,j})^2}{\sum_{k=1}^{N} u_{i,k}} \tag{3.10}$$

Při projekci shluků získaných Gustafson-Kesselovým algoritmem je situace jednoduší. Kromě středu máme také deformační matici každého shluku, která určuje tvar i jeho velikost. Poloosy elipsoidních shluků jsou určeny,

$$r_{i,j} = \sqrt{\lambda_{i,j}} \tag{3.11}$$

kde $\lambda_{i,j}$ jsou vlastní čísla deformační matice $\boldsymbol{P}_i$.

Konstanty $\boldsymbol{a}_i$ a b_i můžeme vypočítat pomocí vážené lineární regresní analýzy. Lze použít i metodu celkové regresní analýzy, avšak v tom případě se nám ztratí interpretace lokálního modelu. Pro použití vážené lineární regresní analýzy se budeme snažit minimalizovat výraz

$$\boldsymbol{S}_i = \sum_{k=1}^{N} (y_k - b_i - \boldsymbol{a}_i^T \boldsymbol{x}_k)^2 \boldsymbol{A}_i(\boldsymbol{x}_k) \tag{3.12}$$

Minimalizaci lze také výhodněji přepsat do kompaktnější maticové formy

$$min_{\boldsymbol{\Theta}_i} \frac{1}{N} (\boldsymbol{y} - \boldsymbol{X}_e \boldsymbol{\Theta}_i)^T \boldsymbol{\Phi}_i (\boldsymbol{y} - \boldsymbol{X}_e \boldsymbol{\Theta}_i) \tag{3.13}$$

kde $\boldsymbol{X}_e$ je matice $\boldsymbol{X}$ rozšířená o jednotkový sloupec, $\boldsymbol{\Theta}_i$ je matice parametrů a $\boldsymbol{\Phi}_i$ je diagonální matice podle rovnic (3.14), (3.15) a (3.16),

$$\boldsymbol{X}_e = [\boldsymbol{X}\,\mathbf{1}] \tag{3.14}$$

$$\boldsymbol{\Theta}_i = [\boldsymbol{a}_i^T\, b_i] \tag{3.15}$$

$$\boldsymbol{\Phi}_i = \begin{bmatrix} u_{i,1} & 0 & \cdots & 0 \\ 0 & u_{i,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{i,N} \end{bmatrix} \tag{3.16}$$

kde $u_{i,k} = A_{i,j}(x_{k,j})$. Výsledek řešení minimalizace (3.13) vede přes soustavu lineárních rovnic s konečným tvarem

$$\boldsymbol{\Theta}_i = (\boldsymbol{X}_e \boldsymbol{\Phi}_i \boldsymbol{X}_e)^{-1} \boldsymbol{X}_e^T \boldsymbol{\Phi}_i \boldsymbol{y} \tag{3.17}$$

Na závěr této části je ještě dobré dodat, že se často setkáváme s modely, které jsou charakterizovány mnoha veličinami. Vstupní prostor je tedy mnohodimenzionální a pokud možno, je vhodné systém rozdělit tak, že vytvoříme několik menších fuzzy modelů, které potom spojíme opět pomocí soustavy Takagi-Sugeno pravidel do jednoho celku.

4 Aplikace algoritmů

V této kapitole si ukážeme výsledky výše popsaných algoritmů na dvou systémech. První z nich je statický nelineární s jedním vstupem, je zde uveden spíše na ukázku a slouží jako názorný rozbor algoritmů. Druhý je dynamický diskrétní nelineární systém s jedním vstupem. Jak již bylo zmíněno v kapitole 3, u dynamických systémů musíme zavádět zpětnou vazbu v podobě výstupů z předchozích měření, čímž se počet vstupů pro naší analýzu rozrůstá.

Všechna měření byla provedena v prostředí MATLAB® za pomoci Fuzzy Logic Toolboxu, který je poskytován současně s výpočetním nástrojem. Dále bylo využito Fuzzy Clustering Toolboxu, který je volně dostupný. Na CD jsou přiloženy kompletní výsledky, které se do práce nevešly z důvodu jejich rozsahu, a použitý skript, který z výsledků fuzzy analýzy sestaví Takagi-Sugeno modely. Dále je pak jeho pomocí možné otestovat modely různými dalšími vstupy.

Identifikace systémů probíhala následovně. Nejdříve se pro určený typ trénovacích dat pozorovala odezva systému. Poté se shromážděná data rozdělila pomocí fuzzy shlukových algoritmů do skupin, ze kterých se vytvořily lokální modely systému a z nich se následně sestavil i globální model. Chování modelů bylo dále zkoumáno na vstupních signálech Pro první model posloužily vstupy typu rampy. Více druhů signálu nebylo nutno testovat, protože se ve statickém systému odezva na stejnou hodnotu vstupu nemění. Druhý systém jsme prověřili vstupy typu jednotkový skok, rampa, sinus a náhodná data. O tom, jak se hodnotila kvalita sestavených modelů pojednává další část.

4.1 Porovnání kvality modelů

Když jsou sestaveny modely systému, je nutné je nějakým způsobem porovnat. Různé modely se porovnávaly ze dvou hledisek.

Prvním bylo kvalita samotného modelu. Pro sadu stejných vstupních dat lze různé modely porovnat, jak moc se lišily od příslušných systémů. K porovnání bylo využito střední kvadratické odchylky,

$$RMSE = \frac{1}{N}\sum_{k=1}^{N}\left(y^{A}(\boldsymbol{x}_k) - y(\boldsymbol{x}_k)\right)^2 \tag{4.1}$$

kde $y^A(\boldsymbol{x}_k)$ je odezva sestaveného modelu, která by se měla co nejvíce blížit výstupu zkoumaného systému $y(\boldsymbol{x}_k)$. Tímto jsme získali porovnání modelů na určitou sadu vstupů. Pro statický systém byla standardní kvadratická odchylka postačující. Protože typů vstupů pro dynamický systém bylo více, byl zaveden index kvality modelu I_k, který je prostým součtem středních kvadratických odchylek pro jednotlivé vstupy. Aby tento index mohl být objektivní, bylo nutné vstupy normalizovat. Čím menší střední kvadratická odchylka nebo index kvality, tím lépe model vystihoval chování systému.

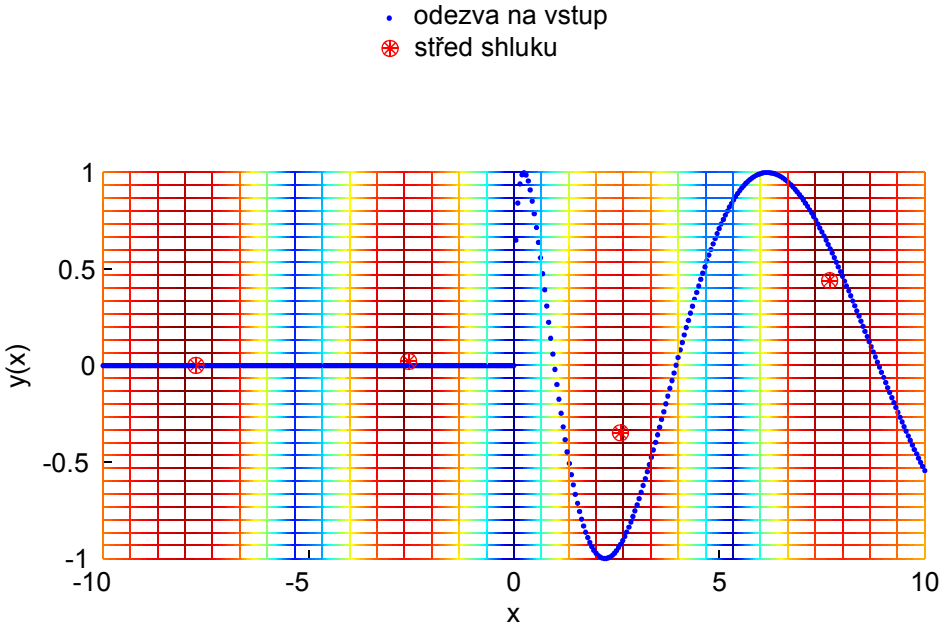
Druhým hlediskem při hodnocení byla časová náročnost sestavení modelu t. Pro úplnost je nutné uvést, že všechny výpočty probíhaly na PC v sestavě s procesorem AMD Duron 1,2GHz, operační pamětí 640 MB RAM a operačním systémem Windows XP.

4.2 Analýza statického systému

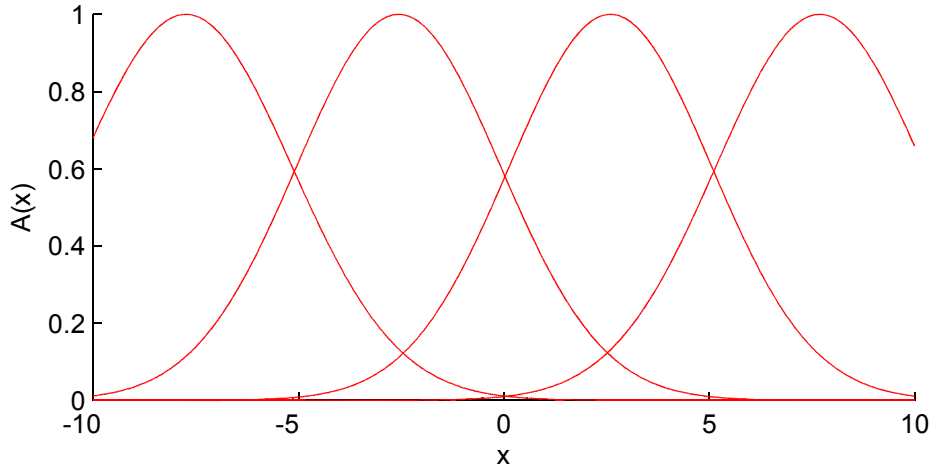
První systém, na kterém byly zkoumány fuzzy shlukové algoritmy, popisuje rovnice

$$\begin{aligned} y(x) &= 0, \quad \text{pro}\, x < 0 \\ y(x) &= \sin(\sqrt{(x)}), \quad \text{pro}\, x \geq 0 \end{aligned} \tag{4.2}$$

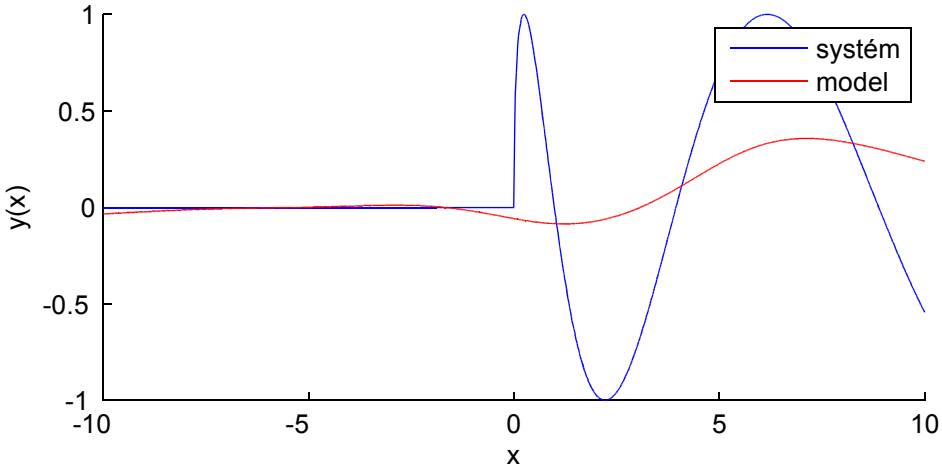
Jedná se tedy o statický nelineární systém s jedním vstupem. Jako trénovací množina byla zvolena množina 400 hodnot funkce typu rampa. Pro tyto vstupy se hledaly shluky v počtu 2 až 20. Na obrázcích s rozložením stupňů příslušnosti není pro přehlednost uvedena legenda. Platí pro ně následující značení



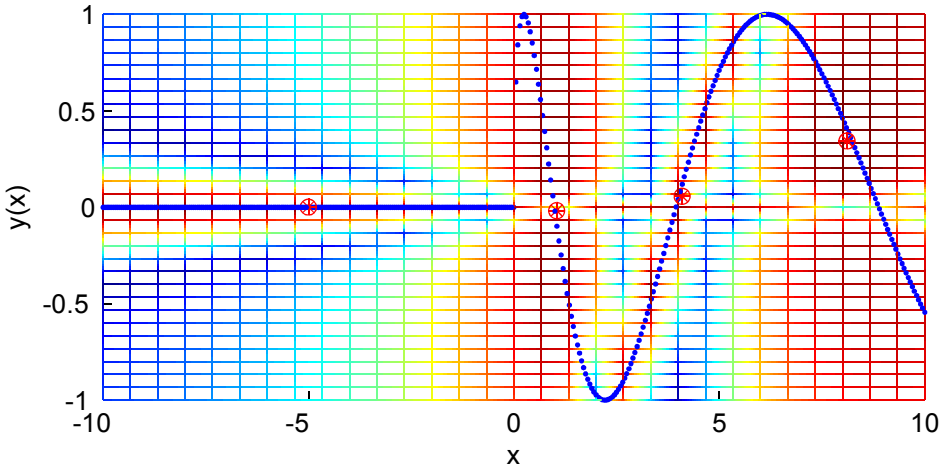
obr 4.1: Fuzzy c-means pro 4 shluky, rozložení stupňů příslušnosti



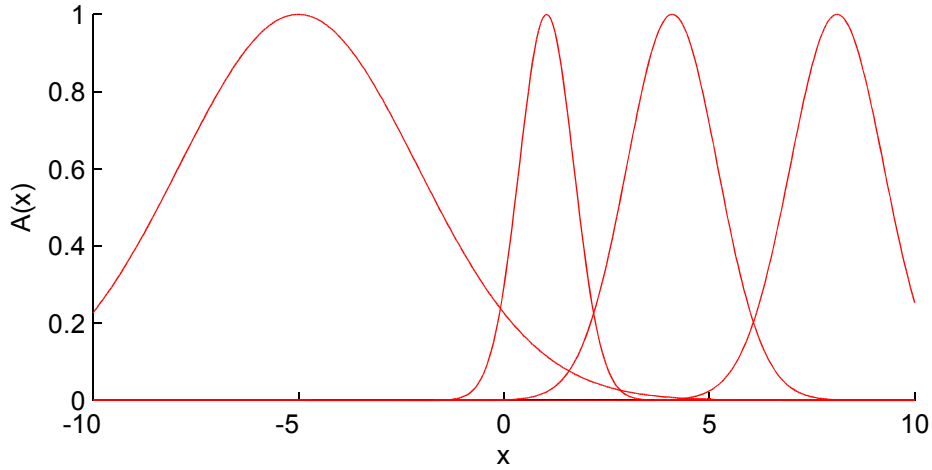
obr 4.2: Fuzzy c-means pro 4 shluky, funkce příslušnosti



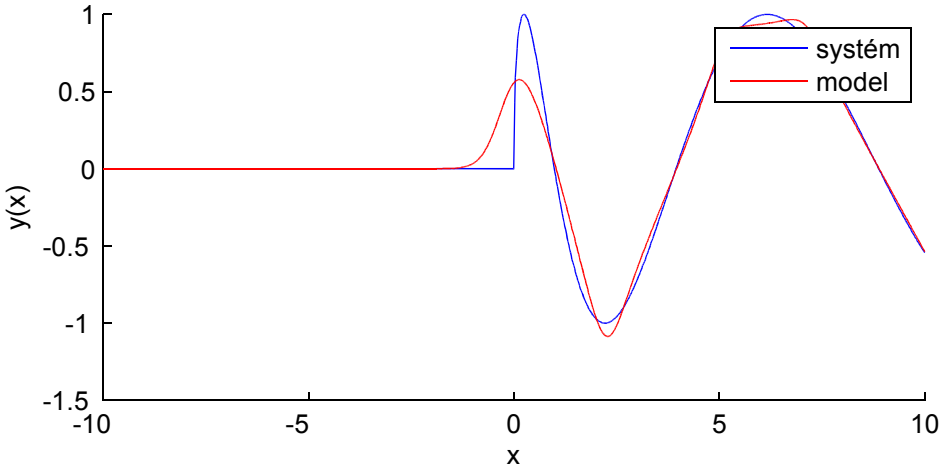
obr 4.3: Fuzzy c-means pro 4 shluky, reakce systému a modelu na rampu, *RMSE* = 0,1636, t = 0,18 s



obr 4.4: Gustafson-Kessel pro 4 shluky, rozložení stupňů příslušnosti

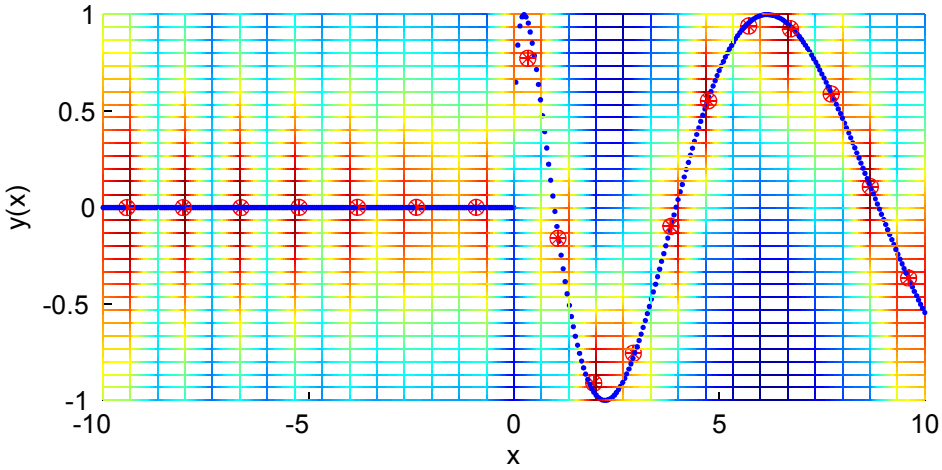


obr 4.5: Gustafson-Kessel pro 4 shluky, funkce příslušnosti

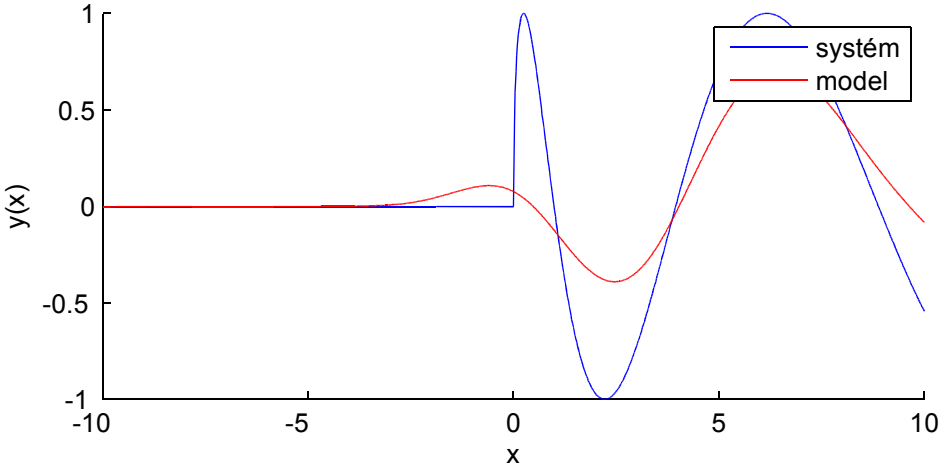


obr 4.6: Gustafson-Kessel pro 4 shluky, odezva systému na rampu, *RMSE* = 0,0154, t = 0,31 s

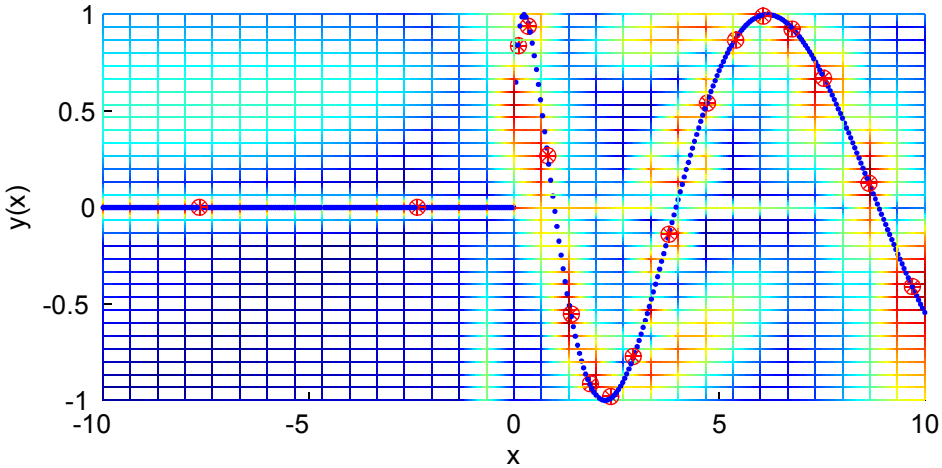
Na obrázcích 4.1 až 4.6 vidíme jak si zkoumané algoritmy pro 4 shluky poradily se systémem a jak reagoval sestavený model. Gustafson-Kesselův algoritmus jednoznačně vracel lepší výsledky díky tomu, že může rozpoznávat shluky různých tvarů a orientací. Funkce příslušnosti mohly tedy být oproti fuzzy c-means roztažené podle příslušných shluků, jak ukazuje obrázek 4.5. Pro oba algoritmy nebyly ovšem modely sestavené ze 4 shluků příliš použitelné. Nejlepší výsledky podávaly pro fuzzy c-means modely sestavené z 18 shluků a pro Gustafson-Kesselův algoritmus modely sestavené z 17 shluků.



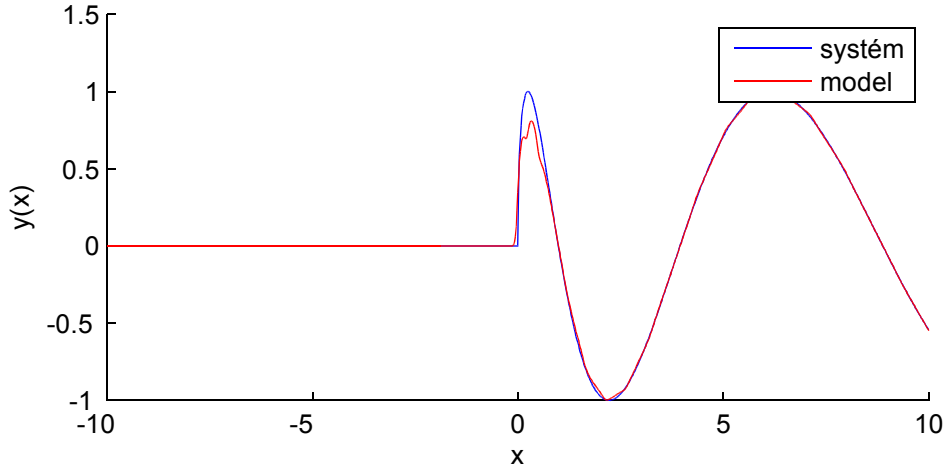
obr 4.7: Fuzzy c-means pro 18 shluků, rozložení stupňů příslušnosti



obr 4.8: Fuzzy c-means pro 18 shluků, odezva na rampu, *RMSE* = 0,069, *t* = 4,5 s

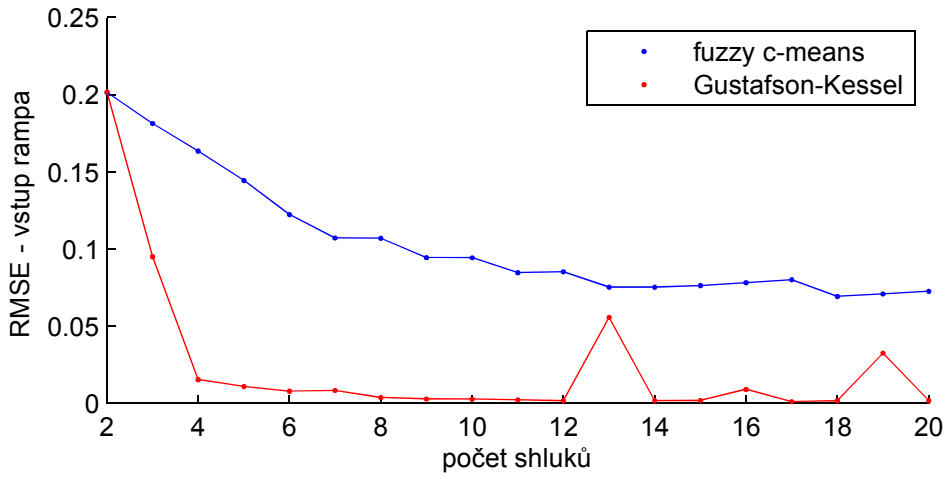


obr 4.9: Gustafson-Kessel pro 17 shluků, rozložení stupňů příslušnosti

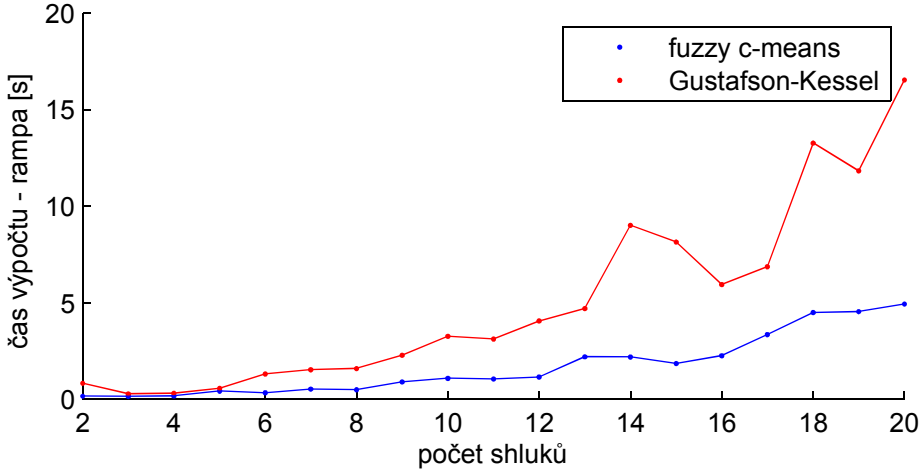


obr 4.10: Gustafson-Kessel pro 17 shluků, odezva na rampu, *RMSE* = 0,0013, *t* = 6,87 s

I přes vysoký počet pravidel (shluků) se algoritmus fuzzy c-means neosvědčil při aproximaci zkoumaného systému. Naopak Gustafson-Kesselův algoritmus prokazoval celkem uspokojivé výsledky. Větší odchylky od správného výstupu vykazoval model v situaci, kdy rychle přecházel velký rozdíl hodnot. Jak je patrno z obrázku 4.10, je to například pro $x \in (0,1)$. Následující dva grafy shrnují kvality modelů a časovou náročnost algoritmů pro jejich sestavení. Pro přehlednost jsou výsledky jednotlivých měření proloženy přímkou.



obr 4.11: Porovnání kvality modelů pro jednotlivé algoritmy



obr 4.12: Porovnání časové náročnosti výpočtu modelů pro jednotlivé algoritmy

4.3 Analýza dynamického systému

Druhým zkoumaným systémem jsme zvolili dynamický diskrétní nelineární systém s jedním vstupem z [8] popsaný následující rovnicí

$$y(k)=f(y(k-1),y(k-2))+x(k) \tag{4.3}$$

kde

$$f(y(k-1),y(k-2))=\frac{y(k-1)y(k-2)[y(k-1)-0{,}5]}{1+y(k-1)^2+y(k-2)^2} \tag{4.4}$$

Jako trénovací množinu dat pro sestavování modelů jsme použily následující typy signálů:

- rampa – 400 vzorků
- sinus – 400 vzorků
- náhodná data – 400 vzorků
- mix třech předchozích – 1200 vzorků

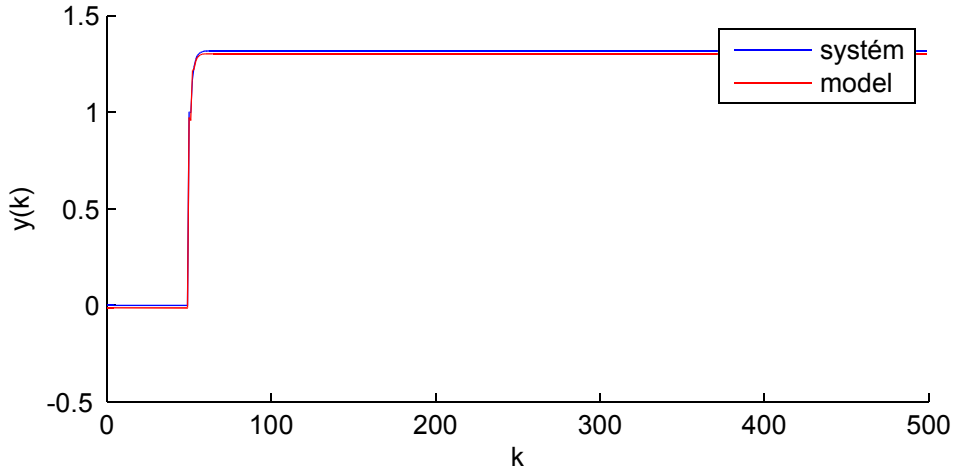
Dále byl index kvality modelu testován pro každý druh vstupu na následujících výstupech:

- jednotkový skok
- rampa
- sinus

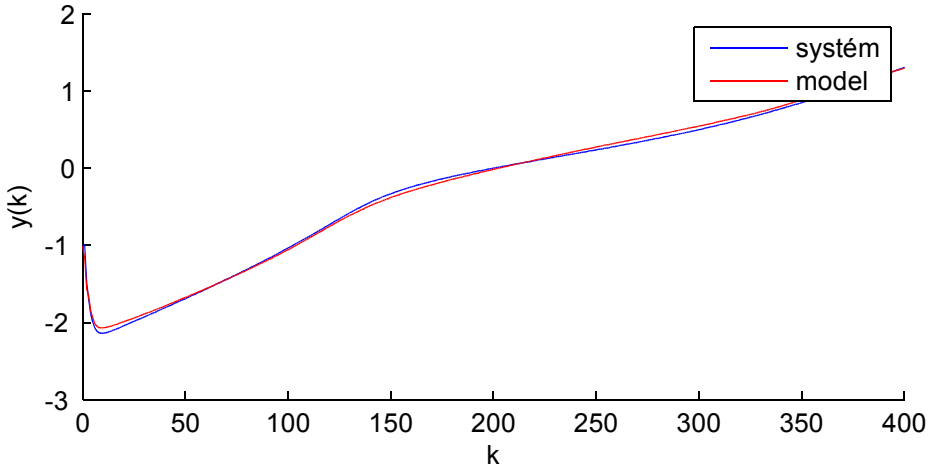
- náhodná data

Následující obrázky ukazují chování třech nejlepších modelů na testované vstupy. Jedná se o Takagi-Sugeno modely sestavené pomocí algoritmu:

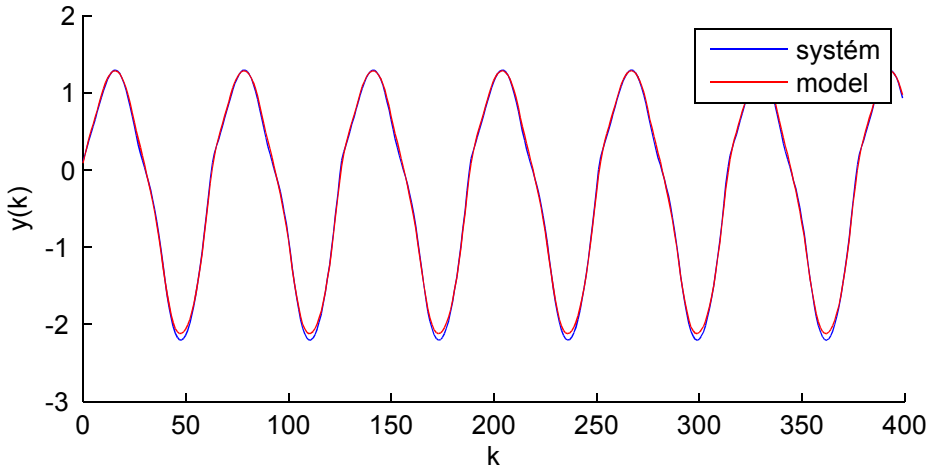
- fuzzy c-means pro 6 shluků trénované na smíšené vstupy, $I_k = 0{,}0091$, $t = 2{,}41$s
- fuzzy c-means pro 20 shluků trénované na vstup typu rampy, $I_k = 0{,}0260$, $t = 10$s
- Gustafson-Kessel pro 3 shluky trénované na smíšené vstupy, $I_k = 0{,}0058$, $t = 1{,}95$s



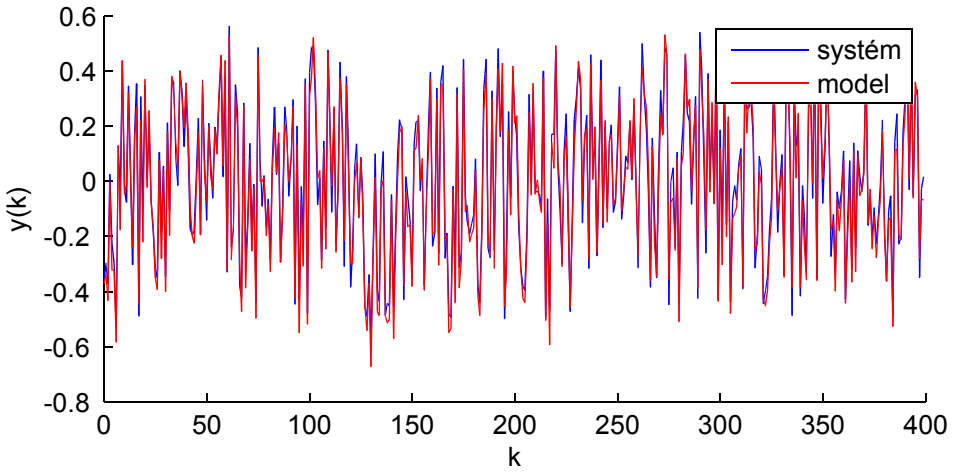
obr 4.13: Fuzzy c-means pro 6 shluků, smíšená data jako trénovací množina, odezva na jednotkový skok, *RMSE* = 0,0002



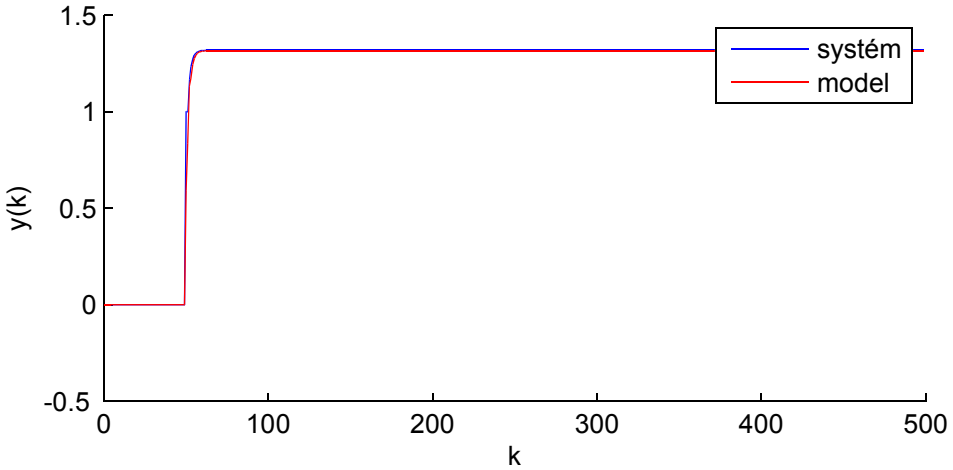
obr 4.14: Fuzzy c-means pro 6 shluků, smíšená data jako trénovací množina, odezva na rampu, *RMSE* = 0,0016



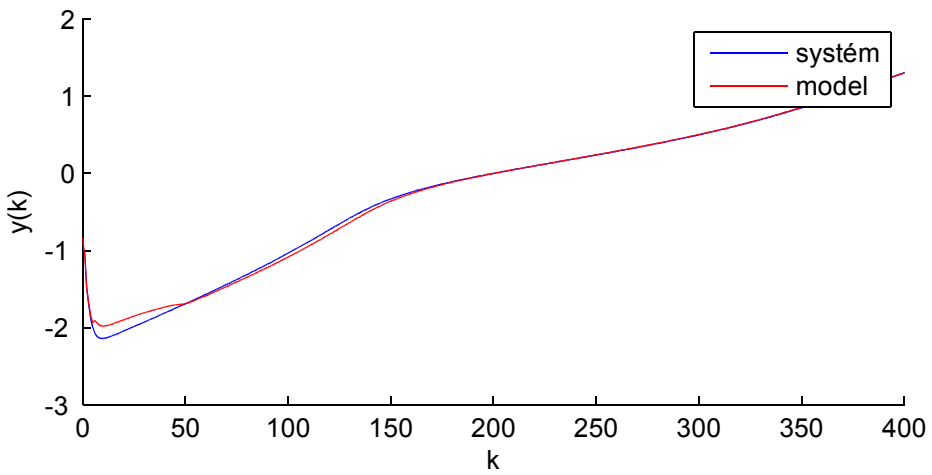
obr 4.15: Fuzzy c-means pro 6 shluků, smíšená data jako trénovací množina, odezva na sinus, *RMSE* = 0,0018



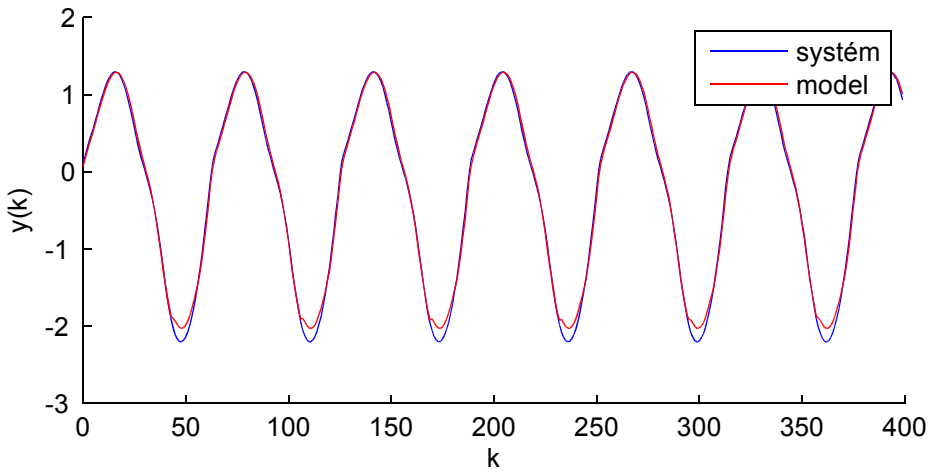
obr 4.16: Fuzzy c-means pro 6 shluků, smíšená data jako trénovací množina, odezva na náhodná data, *RMSE* = 0,0028



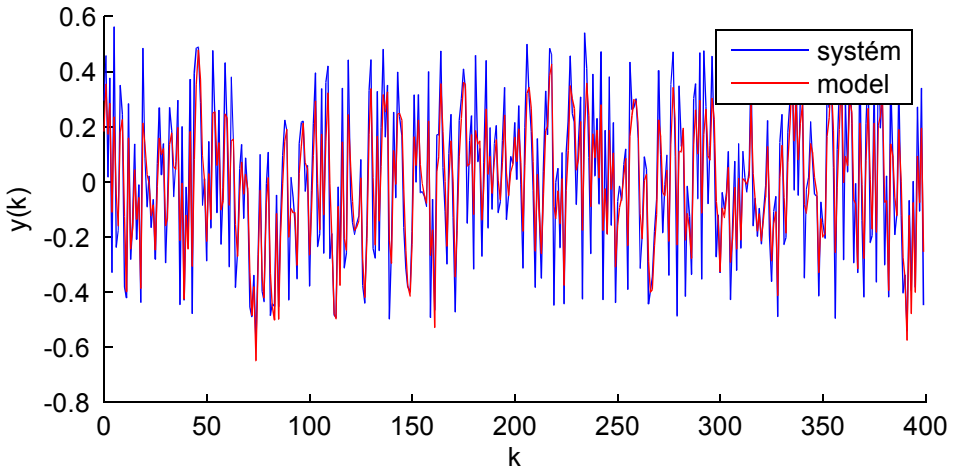
obr 4.17: Fuzzy c-means pro 20 shluků, trénovací množina typu rampa, odezva na jednotkový skok , *RMSE* = 0,0001



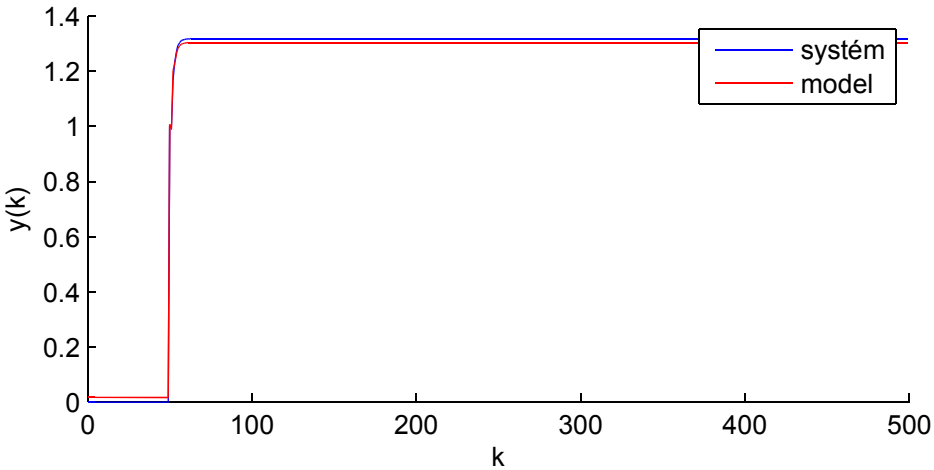
obr 4.18: Fuzzy c-means pro 20 shluků, trénovací množina typu rampa, odezva na rampu, *RMSE* = 0,0023



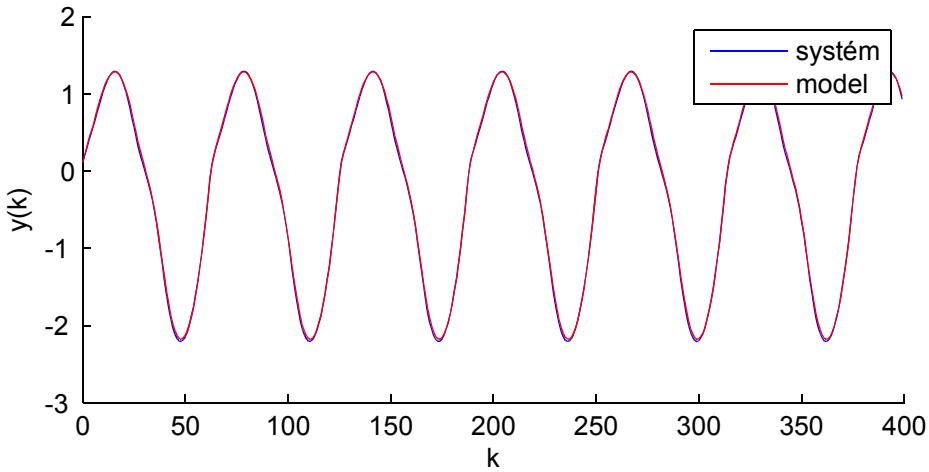
obr 4.19: Fuzzy c-means pro 20 shluků, trénovací množina typu rampa, odezva na sinus, *RMSE* = 0,0059



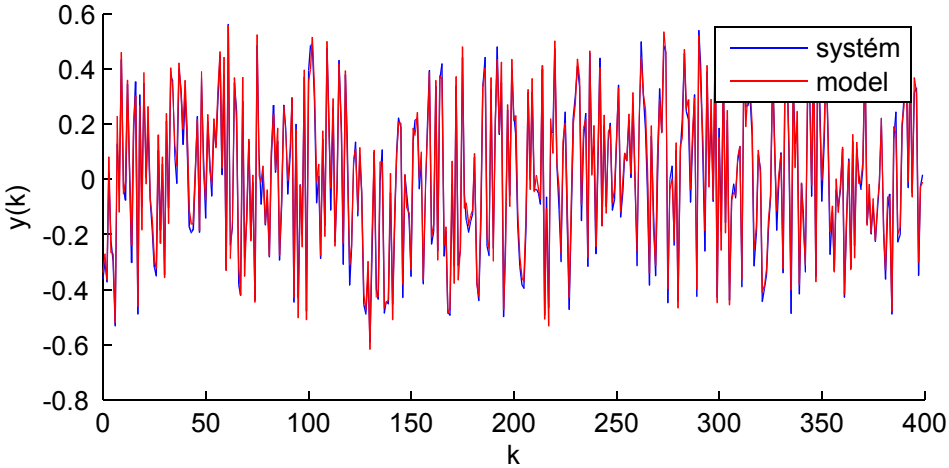
obr 4.20: Fuzzy c-means pro 20 shluků, trénovací množina typu rampa, odezva na náhodná data, *RMSE* = 0,0154



obr 4.21: Gustafson-Kessel pro 3 shluky, smíšená data jako trénovací množina, odezva na jednotkový skok, *RMSE* = 0,0003

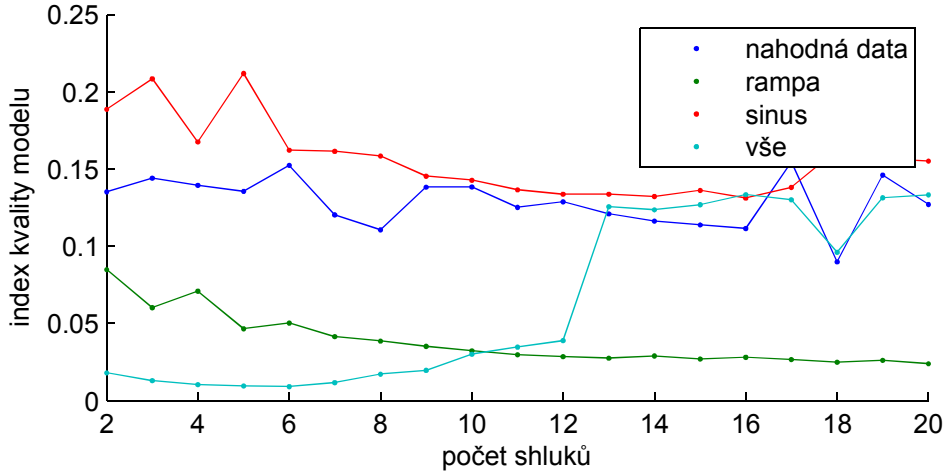


obr 4.22: Gustafson-Kessel pro 3 shluky, smíšená data jako trénovací množina, odezva na sinus, *RMSE* = 0,0010

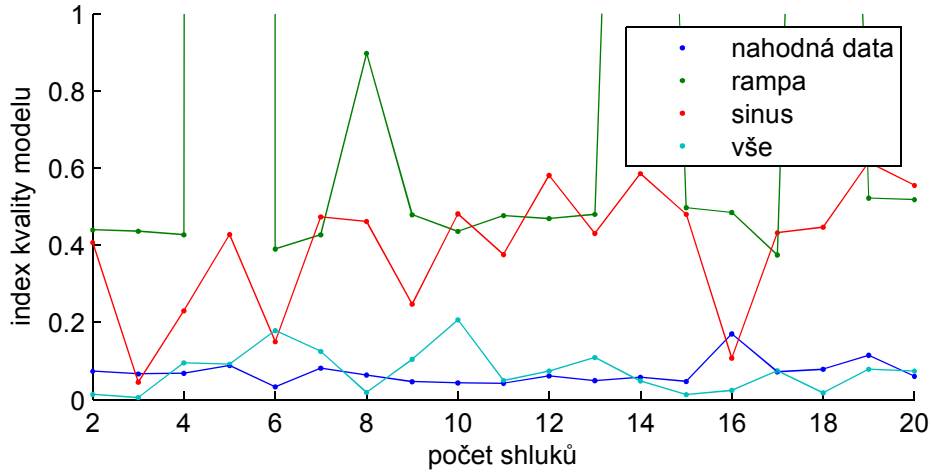


obr 4.23: Gustafson-Kessel pro 3 shluky, smíšená data jako trénovací množina, odezva na náhodná data, *RMSE* = 0,0015

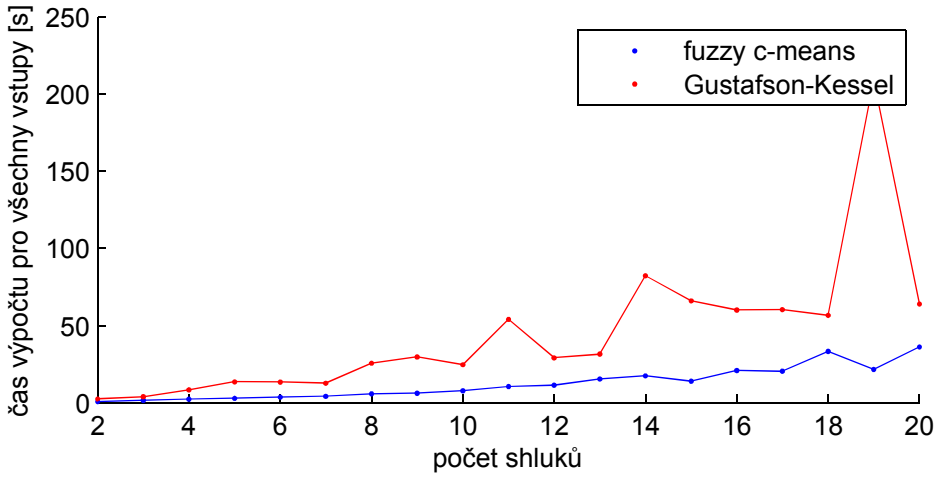
Pro přehled je dále uvedeno porovnání všech modelů, jak z hlediska indexu kvality, tak z hlediska časové náročnosti pro sestavení.



obr 4.24: Kvalita modelů sestavených pomocí fuzzy c-means trénovaných na různé vstupy



obr 4.25: Kvalita modelů sestavených pomocí Gustafson-Kesselova algoritmu trénovaných na různé vstupy



obr 4.26: Porovnání časové náročnosti při sestavování fuzzy modelů

5 Závěr

V této práci byly ukázány principy fuzzy shlukové analýzy fuzzy c-means a Gustafson-Kesselova algoritmu. Následně jsme je použili k sestavování Takagi-Sugeno modelů, které se snažily co nejpřesněji kopírovat chování nelineárních systémů. Zmíněné fuzzy modely byly aplikovány na statický a dynamický nelineární systém.

Pro daný statický systém jsme se přesvědčili, že se návrh fuzzy modelů za pomoci fuzzy c-means neosvědčil. I při vysokém počtu pravidel se podle obr. 4.8 a 4.11 ukázalo, že tyto modely nejsou schopny dostatečně věrně aproximovat systém. Naopak Gustafson-Kesselův postup se jevil jako vhodný, kdy nejlepších výsledků dosahoval při 17 shlucích. Z hlediska časové náročnosti vyšel z testů lépe fuzzy c-means, protože při procesu hledání shluků nemusel počítat inverzi kovarianční matice jako v případě Gustafson-Kesselova algoritmu.

Pro nelineární systém situace nebyla tak zcela jasná jako u prvního případu. Nejlepších výsledků sice dosáhl Takagi-Sugeno model sestavený Gustafson-Kesselovým algoritmem pro 3 shluky a pro smíšené vstupy obr. 4.21 až 4.23, ale celkově si vedly lépe modely sestavené s pomocí fuzzy c-means. Podle očekávání obdobně jako u prvního testovaného systému, modely získané fuzzy c-means analýzou byly časově méně náročné na sestavení.

Některé výsledky analýzy jsou ovšem ovlivněny chybou v použitém Fuzzy Clustering Toolboxu, potažmo prostředí MATLAB® 7.4, kdy při počítání kovarianční matice vznikaly velmi malé hodnoty a funkce pro inverzi matice si s nimi nedokázala poradit. Velmi dobře je chyba vidět na obr. 4.25, kdy pro náhodná data a 5 shluků je index kvality řádově větší než ostatní, právě v důsledky této chyby. Tento nedostatek lze odstranit zavedením přesnosti $\varepsilon > 0$. Pokud by se v kovarianční matici vyskytlo číslo menší než ε, nastavila by se jeho hodnota na ε. Tato oprava byla testována pro $\varepsilon = 10^{-5}$ a je funkční, nebyla už ovšem zahrnuta do výsledků.

Zjistili jsme, že Takagi-Sugeno modely jsou použitelné pro modelování nelineárních systémů, i když by bylo dobré vyzkoušet je na více systémech. Porovnání dalších fuzzy shlukových algoritmů jako Gath-Geva by přineslo další přehled o použitelnosti těchto fuzzy modelů. Zajímavé výsledky bychom získali i porovnáním Takagi-Sugeno modelů s jinými typy, jako jsou například neuro-fuzzy modely. Nespornou výhodou Takagi-Sugeno modelů ovšem zůstane jejich jasná interpretace.

Reference

[1] Zadeh L. A., *Information & Control*, 1965,

[2] Hoppner F., Klawonn F., Kruse R., Runkler T., *Fuzzy Cluster Analysis*, 1999, 471-98864-2

[3] Novák V., *Základy fuzzy modelování*, 2003, 80-7300-009-1

[4] Turčan A., Ocelíková E., Madarász L., *Fuzzy c-means Algorithms in Remote Sensing*,

[5] Golob M., *Fuzzy Modeling and Control*, 2000

[6] Štefan M., *Identifikace nelineárních systémů pomocí neuro-fuzzy modelů*,

[7] Potts D., Sammut C., *Online Nonlinear System Identification in High Dimensional Environments*,

[8] Abonyi J., Babuška R., Szeifert F, *Modified Gath-Geva Fuzzy Clustering for Identification of TS Fuzzy Models*,

[9] Abonyi J., Babuška R., Verbruggen H.B., Szeifert F, *Incorporating Prior Knowledge in Fuzzy Model Identification*,

[10] Abonyi J., Balasko B., Feil B., Fuzzy Clustering Toolbox, 2008, http://www.fmt.vein.hu/softcomp/fclusttoolbox/

Obsah CD

\	- kořenový adresář, zdrojový kód dokumentu, tisknutelný dokument v pdf formátu
\obr	- v podsložkách odpovídajícím číslům kapitol jsou uloženy obrázky použité v dokumentu
\matlab	- skripty MATLAB sloužící k analýze
\matlab\dokument	- skripty MATLAB pro generování grafiky do dokumentu
\matlab\vysledky	- výsledky analýzy 1 a 2
\matlab\vysledky\analyza1	- podrobné výsledky analýzy 1
\matlab\vysledky\analyza2	- podrobné výsledky analýzy 2