

LFLC 2000 + MATLAB/SIMULINK - SYSTÉM PRO UNIVERSÁLNÍ APLIKACE FUZZY LOGIKY

Antonín Dvořák, Hashim Habiballa, Vilém Novák a Viktor Pavliška

Abstrakt. Softwarový balík LFLC 2000 je komplexním nástrojem pro návrh jazykových popisů (tj. soustav fuzzy pravidel IF-THEN) určených pro fuzzy regulaci a řízení, rozhodování, popř. klasifikaci. Jeho potenciální využití leží jak v průmyslových, tak v akademických aplikacích. Propojení se systémem MATLAB/Simulink umožňuje dále rozšířit možnosti LFLC 2000.

1 Úvod

Softwarový balík LFLC 2000 je prostředkem pro návrh, ladění a zpracování jazykových popisů, tj. soustav fuzzy pravidel typu JESTLIŽE-PAK. Tato pravidla mají tvar podmínky a vyskytují se v nich vágní výrazy přirozeného jazyka, např. *malý*, *velmi velký*, *zhruba střední* apod. Software je vyvíjen na Ústavu pro výzkum a aplikace fuzzy modelování (ÚVAFM) při Ostravské Univerzitě v Ostravě. Jde o vylepšenou verzi původního programového vybavení LFLC (Linguistic Fuzzy Logic Controller) 1.5, který vyvinul prof. V. Novák v 90. letech minulého století pod platformou MS-DOS. Původní LFLC 1.5 byl využíván nejen v akademickém prostředí, ale i v průmyslových aplikacích. Potřeba zmodernizovat nejen design aplikace, ale i jeho modelovací schopnosti vedly k vývoji jeho verze pro systém WINDOWS, kterou je LFLC 2000. Vzhledem k rozvoji informačních technologií nebyl systém vyvíjen jako samostatná aplikace, ale bylo využito moderních prostředků a komunikačních protokolů s využitím produktu MATLAB. Software je již nyní ve stádiu distribuovatelné verze, která je na ÚVAFM stále zdokonalována, jak po stránce implementační, tak teoretické. Na softwaru spolupracuje tým pracovníků spolu se studenty katedry matematiky Přírodovědecké fakulty Ostravské univerzity. Je nutné zdůraznit, že LFLC 2000 je softwarovou realizací původních vědeckých výsledků dosažených na Ostravské univerzitě, které jsou mezinárodně uznávány, a systém v žádném případě nekopíruje jiné softwarové systémy vyvíjené jinde, a to včetně MATLAB/FuzzyLogic toolbox.

2 Teoretické základy

Aplikace umožňuje navrhovat báze pravidel typu:

JESTLIŽE X_1 je A_1 A ... A X_n je A_n PAK Y je B ,

kde A_i a B jsou jazykové výrazy charakterizující hodnoty proměnných X_i a Y .

Příklad:

Představme si situaci, kdy budeme modelovat akční zásahy řidiče v automobilu na brzdou soustavu vzhledem ke sledovaným veličinám vzdálenosti překážky od vozidla a rychlosti vozidla. Pravidlo pak může vypadat takto:

Jestliže VZDÁLENOST PŘEKÁŽKY je *malá* a RYCHLOST je *velká* **pak** SÍLA PŮSOBÍCÍ NA PEDÁL BRZDY je *velmi velká*.

V tomto případě byly použity speciální výrazy přirozeného jazyka, a to „malý“ a „velmi velký“. Slovo „velmi“ se nazývá jazykový operátor a modifikuje význam výrazu „malý“.

LFLC 2000 umožňuje používat standardní jazykové výrazy i definovat a testovat použití vlastních výrazů (s uživatelsky definovaným významem). Standardní výrazy jsou *malý*,

střední, velký, popř. doplněné o jeden z jazykových operátorů *výrazně, značně, velmi, spíše, více méně, zhruba, dosti zhruba, velmi zhruba*. V aplikacích v oblasti automatického řízení, kde je nutno uvažovat i záporné hodnoty proměnných, můžeme používat i výrazů typu „-*střední*“ apod. Báze takto definovaných pravidel vytvářejí pak jazykový popis chování systému. Aplikace nejen umožňuje návrh takových pravidel a testování jejich chování (viz oddíl 3), ale má také implementovány různé druhy metod pro odvozování hodnot výstupních proměnných v závislosti na vstupních hodnotách (tzv. inferenční neboli dedukční metody). Na tomto místě je vhodné podotknout, že uživatel pracuje *hlavně s jazykovými výrazy* a modifikuje jazykový popis pouze pomocí nich. Počítač se tak stává partnerem, který „rozumí“ přirozenému jazyku a „umí“ na něj reagovat.

Dvě základní inferenční (dedukční) metody, které systém LFLC 2000 obsahuje, jsou fuzzy aproximace (dobře známá z literatury i aplikací jako *Mamdaniho max-min* pravidlo) a *fuzzy logická dedukce*. Tato metoda, soustavně rozvíjená na našem pracovišti, interpretuje JESTLIŽE-PAK pravidla jako *jazykově zadané logické implikace*, a odvozování závěru pak probíhá jako dedukce pomocí inferenčního pravidla *modus ponens*. Přibližná dedukce v LFLC 2000 pracuje na principu zadání vstupních hodnot při určité bázi pravidel a aplikaci inferenční metody, která vytvoří výstupní fuzzy množinu. Tu pak lze defuzzifikovat na výstupní hodnotu. Program umožňuje i výběr vhodné metody defuzzifikace. Pro fuzzy logickou dedukci je zde navržena a implementována speciální defuzzifikační metoda DEE (defuzzifikace evaluačních výrazů), protože standardní běžně používaná metoda COG (těžiště) v tomto případě nefunguje správně.

Součástí systému je také původní metoda, pomocí níž lze automaticky „naučit“ jazykový popis na základě dat a aproximovat data pomocí metod fuzzy logiky.

Systém LFLC 2000 je navržen tak, aby uživatel pracoval zejména s jazykovými popisy. Ladění těchto popisů pak spočívá v modifikaci použitých jazykových výrazů, tj. uživatel mění slova použitá v popisu. Pouze zcela výjimečně je třeba měnit i fuzzy množiny - významy těchto jazykových výrazů. Tento rys odlišuje softwarový systém LFLC 2000 od ostatních softwarových systémů implementujících metody fuzzy logiky. Více informací o teoretických základech fuzzy logiky je možné najít v literatuře uvedené na konci článku, zejména v pracích [5] a [7]. Samozřejmě, standardní postup používaný v jiných systémech je v LFLC 2000 také možný.

3 Aplikace

LFLC umožňuje návrh, testování a učení jazykových popisů. Tyto popisy mohou být využity v řízení, podpoře rozhodování a dalších aplikacích. Můžeme rozlišit tyto základní úkoly realizované pomocí LFLC:

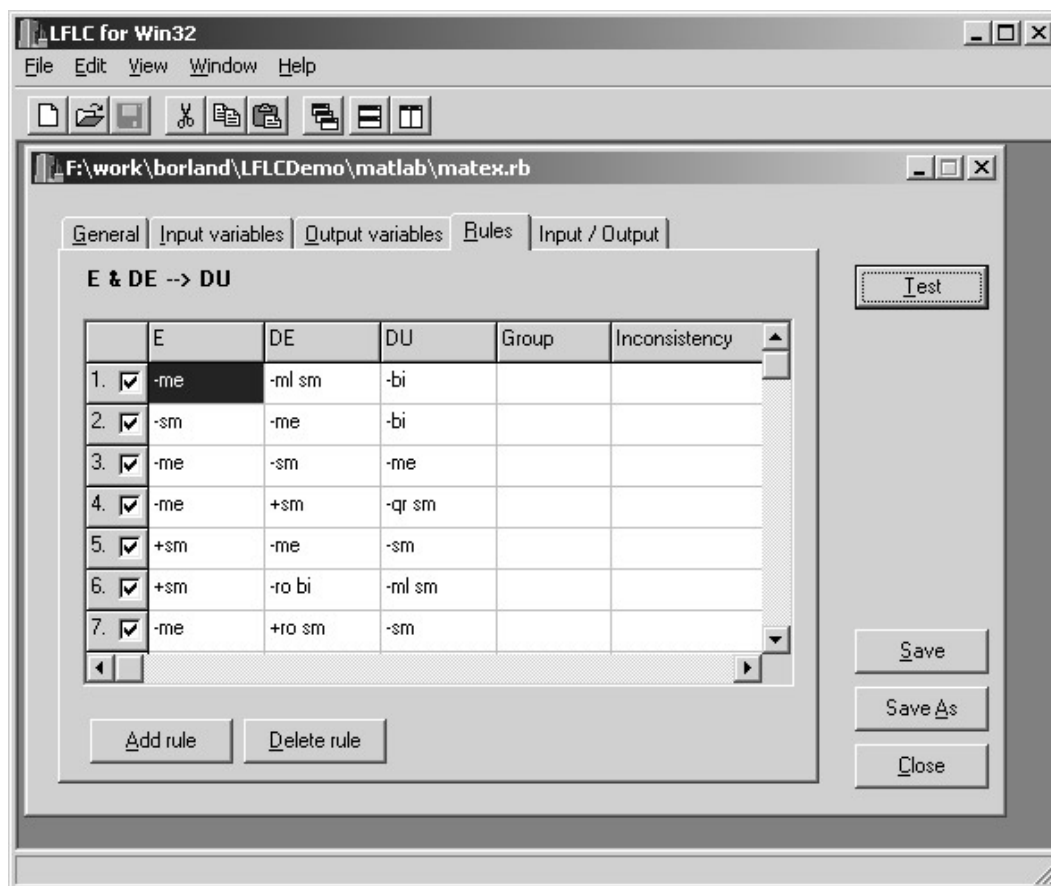
- **návrh jazykových popisů.** LFLC umožňuje použít různé předdefinované jazykové výrazy. Má také prostředky pro analýzu vlastností popisů (detekce identických nebo nekonsistentních pravidel atd.).
- **návrh a modifikace uživatelských výrazů.** K předdefinovaným jazykovým výrazům lze navíc definovat uživatelské výrazy v situacích, kdy jsou standardní nedostatečné.
- **testování inference nad navrženými jazykovými popisy.** LFLC umožňuje uživateli sledovat chování jazykových popisů pro různé hodnoty. Může vybírat různé inferenční a defuzzifikační metody a sledovat projekce proti různým proměnným. Také má k dispozici informace o aktivovaných pravidlech a nejvhodnějších jazykových výrazech přiřazených hodnotám ze vstupních a výstupních intervalů.

- **učení jazykových popisů z experimentálních dat.** LFLC implementuje několik metod učení se z experimentálních dat.
- **modelování regulačních procesů.** LFLC má díky klientovi pro MATLAB (Simulink) možnost vytvořené a otestované jazykové popisy využít k simulaci regulačního procesu na modelech systémů (v Simulinku).

4 Implementace a komunikace s ostatními produkty

LFLC 2000 je vytvořeno v programovacím jazyce C++ s použitím vizuálního nástroje pro návrh grafického rozhraní Borland C++ Builder pro platformu MS-WINDOWS (32-bit). Systém je rozdělen do několika modulů, které umožňují jeho efektivní rozvíjení. Hlavním stavebním kamenem je jádro systému, které implementuje reprezentaci fuzzy množin, proměnných, báze pravidel, inferenčních metod apod. Toto jádro je nezávislé na grafickém rozhraní a jeho algoritmy jsou přístupné i jiným softwarovým systémům ve formě COM (Component object model) objektu RBaseCOM. Jádro je tvořeno pomocí objektově orientované metodologie.

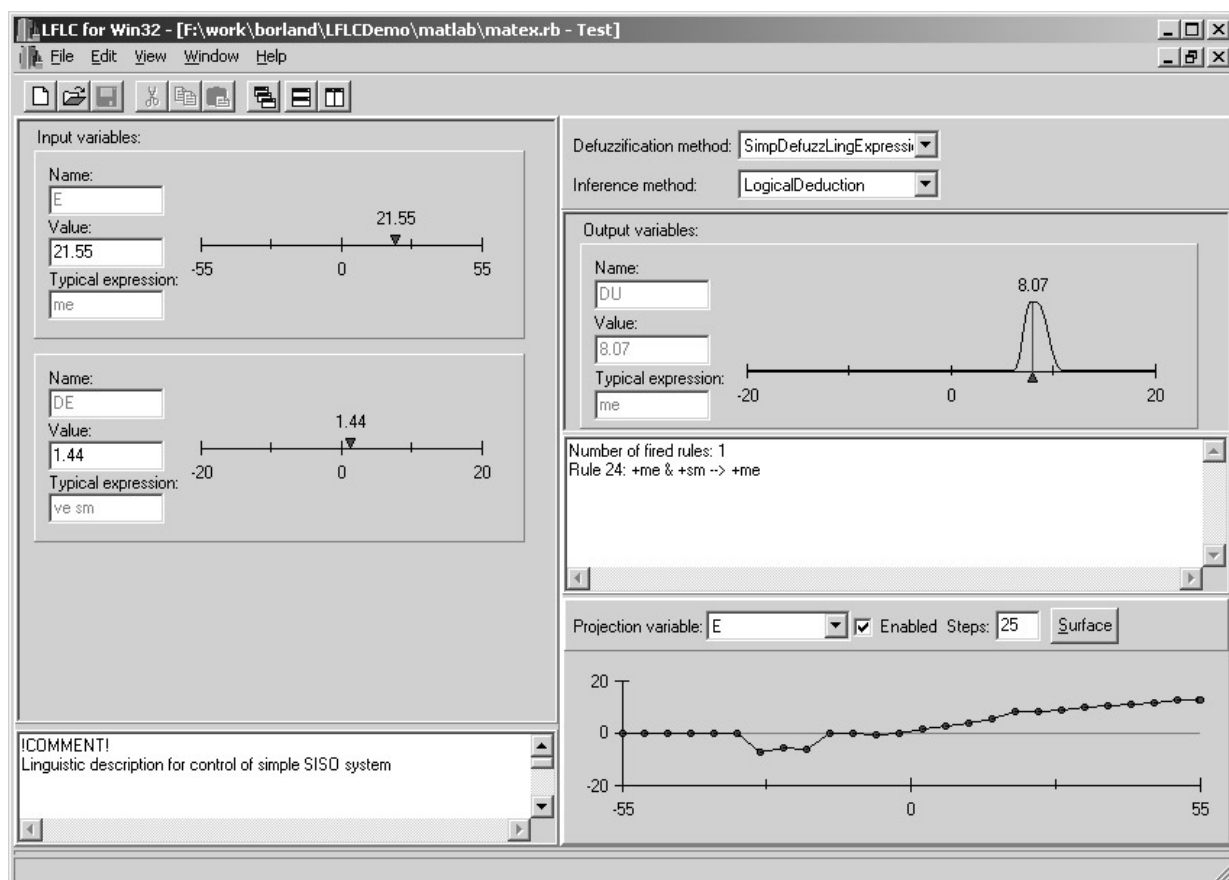
Objekty jádra využívá grafické rozhraní aplikace (GUI). Ukázkou GUI můžete vidět na obrázku 1, který ukazuje obrazovku pro návrh pravidel a sledování jejich vlastností.



Obrázek 1: Návrh pravidel

Sloupce Group a Inconsistency slouží k detekci shodných pravidel, respektive pravidel se stejnou levou a různou pravou stranou. Obrázek 2 ukazuje testovací obrazovku, kde uživatel může zkoumat chování navrženého jazykového popisu. Na levé straně může interaktivně vkládat hodnoty pro vstupní proměnné a na pravé straně může sledovat změny výstupní fuzzy množiny. Zároveň lze vybírat metody inference i defuzzifikace. Také lze

sledovat použitá JESTLIŽE-PAK pravidla a v pravém dolním rohu je znázorněna projekce výstupní proměnné vzhledem k vybrané vstupní proměnné.



Obrázek 2: Testovací obrazovka

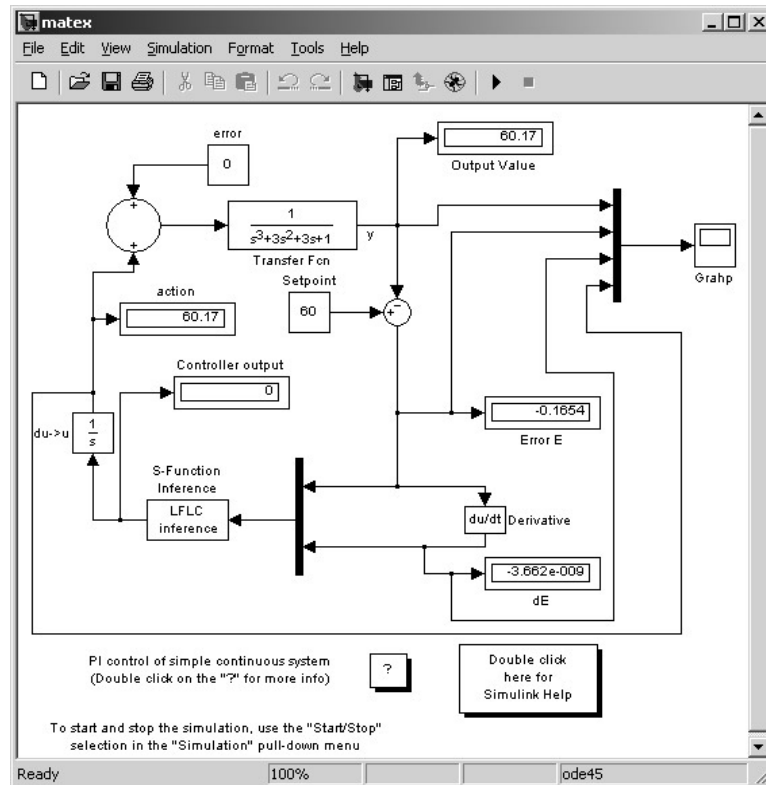
Během návrhu jazykového popisu se musí specifikovat použité proměnné. U nich zpravidla stačí zadat pouze rozsah možných hodnot (v terminologii 2000 hovoříme o jejich *jazykovém kontextu*). Pokud to však uživatel potřebuje, může modifikovat významy používaných standardních jazykových výrazů (tj. mírně upravovat tvar implementovaných fuzzy množin), popř. doplnit své vlastní jazykové výrazy a ty pak používat v popisu.

Z hlediska modelování regulačních systémů je nejdůležitější vytvoření COM objektu, který zapouzdřuje základní metody jádra – zavedení popisu, práci s proměnnými a inferenci nad vstupními daty. Pro práci s MATLABem je vytvořena S-funkce, která umožňuje simulaci různých procesů a jejich řízení.

V prostředí produktu MATLAB lze používat dva přístupy pro práci s algoritmy jádra LFLC 2000 a to buď s použitím již zmiňované S-funkce přímo v simulačních schématech programové nadstavby Simulink, anebo lze používat mex-funkci (LFLCInfer) v uživatelsky zkonstruovaných programech. Tato funkce je naprogramovaná v jazyce C++ a opět využívá dříve zmíněný COM objekt. LFLCInfer umožňuje provádění inferencí z popisů vytvořených v samostatné aplikaci LFLC 2000.

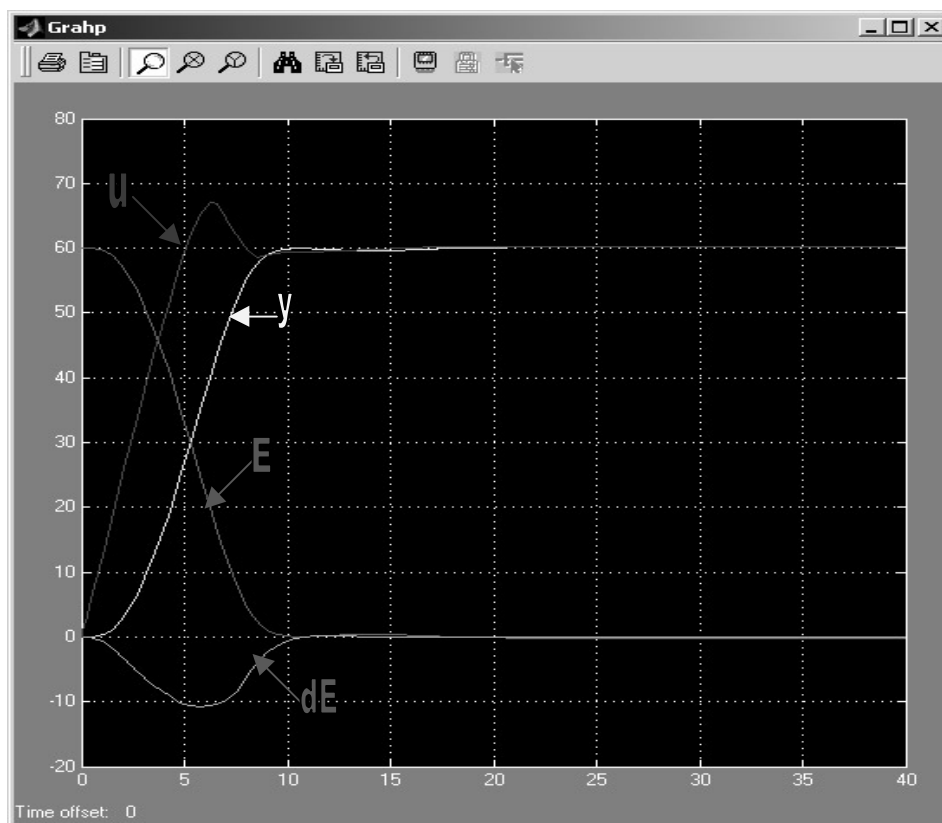
Jako příklad propojení systémů LFLC 2000 a Simulink uvádíme regulaci jednoduchého dynamického SISO systému s přenosovou funkcí $F(p) = \frac{1}{p^3 + 3p^2 + 3p + 1}$. Pro regulaci je

použit jazykový popis matex.rb, jehož část je zobrazena na Obrázku 1. Regulační schéma v systému Simulink znázorňuje Obrázek 3:



Obrázek 3: Regulační schéma

Na obrázku 4 vidíme průběh regulace dynamického systému pomocí jazykového popisu matex.rb.



Obrázek 4: Průběh regulace

Z našich zkušeností se systémem LFLC vyplývá, že pomocí *jednoho* jazykového popisu (s případnými drobnými modifikacemi) je možné úspěšně regulovat *celou řadu dynamických systémů různých řádů i charakteristik*. Přitom jsou zachovány všechny významné vlastnosti fuzzy regulace, zejména značná robustnost - necitlivost vůči změnám podmínek či vstupů a poruchám. Práce s jazykovými popisy je velmi jednoduchá a intuitivní a případné pozdější změny či úpravy nepřinášejí velké problémy, protože i po delší době lze rychle pochopit, co vlastně jazykový popis dělá. To demonstruje velkou sílu a potenciál našeho systému, který takto důsledně respektuje původní záměr fuzzy logiky – umožnit pracovat s přirozeným jazykem a naučit počítače, aby mu rozuměli.

5 Závěr

LFLC 2000 nabízí potenciálně velmi široké použití v celé řadě aplikací, a to jak v regulaci a řízení, tak při různých rozhodovacích problémech nebo při klasifikaci. Jde o systém, který je již v současné době schopen nasazení a bude i v budoucnosti rozvíjen. Jelikož se na systému pracuje současně s vývojem nových teoretických metod, dochází i k jejich rychlé integraci do LFLC 2000 a tím se jeho možnosti ještě dále rozšiřují. Propojení LFLC 2000 se systémem MATLAB/Simulink dále rozšiřuje jeho možnosti.

6 Literatura

1. Bělohávek, R., Dvořák, A., Jedelský, D., Novák, V.: Object Oriented Implementation of Fuzzy Logic Systems. Obsaženo v Camarinha-Matos, L.M. et al. (eds.): *Intelligent Systems for Manufacturing. Multi-Agent Systems and Virtual Organizations*. Kluwer, Dordrecht 1998, 589-594.

2. Dvořák, A., Jedelský, D.: Defuzzification and Chaining of Rules in Hierarchical Fuzzy Systems. Ve sborníku 1999 *Eusflat-Estylf Joint Conference*, 199-202.
3. Dvořák, A, Novák, V.: Fuzzy Logic Deduction with Crisp Observations. Zasláno do *Soft Computing*.
4. Novák, V.: Linguistically Oriented Fuzzy Logic Controller and Its Design. *Int. J. of Approximate Reasoning* **12**(1995), 263-277.
5. Novák, V.: *Základy fuzzy modelování*. BEN, Praha 2000.
6. Novák V., Kovář, J.: Linguistic IF-THEN Rules in Large Scale Application of Fuzzy Control. Obsaženo v Da Ruan, Kerre E.E. (eds) *Fuzzy If-Then Rules in Computational Intelligence: Theory and Applications*. Kluwer, Boston 2000, 223-241.
7. Novák, V., Perfilieva, I., Močkoř, J.: *Mathematical Principles of Fuzzy Logic*. Kluwer, Boston 1999.

Kontakt:

Ústav pro výzkum a aplikace fuzzy modelování, Ostravská universita

30.dubna 22, 701 03 Ostrava 1, Česká Republika

tel.: +420596160234, e-mail: vilem.novak@osu.cz www: <http://ac030.osu.cz/irafm>