

Softwarová podpora pro vytváření fuzzy modelů

Petr Želasko

Katedra měřicí a řídicí techniky, FEI, VŠB – Technická univerzita Ostrava
17. listopadu 15, 708 33, Ostrava-Poruba
petr.zelasko@vsb.cz

Abstrakt. Referát je zaměřen na problematiku softwarové podpory pro vytváření a vyhodnocování fuzzy modelů. Předmětem referátu je knihovna procedur, realizovaných v programovacím jazyce C. Tyto procedury umožňují v programech vytvořených v programovacím jazyce C sestavení fuzzy modelů se vstupy, výstupy a znalostními bázemi, včetně jejich editace, úprav a vyhodnocování výsledků. Knihovna procedur dodržuje normu ANSI C, což zajišťuje její plnou přenositelnost mezi překladači jazyka C (dodržujících normu ANSI C) různých mikroprocesorů.

Klíčová slova: fuzzy logika, fuzzy procedury, implementace, testování

1 Úvod

U mnoha procesorů, podporujících implementaci fuzzy systémů pomocí speciálních instrukcí (tj. fuzzy instrukcí), se setkáváme s problémem, kdy nám tyto vestavěné funkce procesoru sice umožňují sestavování fuzzy systémů, ale jen ve velmi zjednodušené formě. V tomto referátu si jako příklad uvedme procesory řady HC12 firmy Motorola.

2 Nevýhody vestavěných fuzzy instrukcí procesorů řady HC12

Tvar vstupních hodnot lze definovat pouze pomocí lichoběžníkového průběhu. Fuzzy proměnná může nabývat pouze hodnot z celočíselného intervalu $\langle 0, 255 \rangle$. Funkce příslušnosti může nabývat také pouze hodnot z celočíselného intervalu $\langle 0, 255 \rangle$. Výstupní fuzzy proměnná může nabývat pouze hodnot ve tvaru singletonů. Pro defuzzifikaci je v procesoru HC12 implementována pouze jedna metoda (a to metoda váženého průměru (weighted average), která je blíže popsána v literatuře [1]). Pro zadání dotazu je možno použít pouze ostré číslo. Instrukce podporují pouze Mamdaniho model, nikoliv však model Takagi-Sugeno.

3 Knihovna fuzzy procedur „fuzzylib“ (fuzzy library)

Toto byty tedy problémy a nedostatky vestavěných instrukcí procesoru HC12, které do značné míry omezují možnosti tohoto procesoru pracovat s mnoha rozmanitými

druhy fuzzy systémů a omezují se tak jen na určitou malou skupinu těchto systémů. Z tohoto důvodu bylo třeba vytvořit prostředek, který by možnost práce s fuzzy systémy rozšířil a umožnil tak procesoru HC12 jeho širší použití v dané oblasti. Tímto prostředkem se stala knihovna fuzzy procedur s názvem „fuzzylib“ (FUZZY LIBRARY).

3.1 Seznam všech procedur knihovny „fuzzylib“

Procedury sloužící k editaci vstupních a výstupních proměnných:

- 1) Procedura AddFP
- 2) Procedura AddFPInValueT
- 3) Procedura AddFPInValueE
- 4) Procedura ModifyFPInValueT
- 5) Procedura ModifyFPInValueE
- 6) Procedura DeleteFPInValue
- 7) Procedura DeleteFPInValuesChain
- 8) Procedura DeleteFPIn
- 9) Procedura GetFPInValueT
- 10) Procedura GetFPInValueE
- 11) Procedura GetFPInLength
- 12) Procedura GetFPInValuesChainLength
- 13) Procedura AddFPOutValueT
- 14) Procedura AddFPOutValueE
- 15) Procedura AddFPOutCChain
- 16) Procedura AddFPOutValueC
- 17) Procedura ModifyFPOutValueT
- 18) Procedura ModifyFPOutValueE
- 19) Procedura ModifyFPOutValueC
- 20) Procedura DeleteFPOutValueC
- 21) Procedura DeleteFPOutCChain
- 22) Procedura DeleteFPOutValue
- 23) Procedura DeleteFPOutValuesChain
- 24) Procedura DeleteFPOut
- 25) Procedura GetFPOutValueT
- 26) Procedura GetFPOutValueE
- 27) Procedura GetFPOutValueC
- 28) Procedura GetFPOutLength
- 29) Procedura GetFPOutValuesChainLength

Procedury sloužící k editaci báze pravidel:

- 30) Procedura AddRule
- 31) Procedura AddRuleValue
- 32) Procedura ModifyRuleValue
- 33) Procedura DeleteRuleValue
- 34) Procedura DeleteRuleValuesChain
- 35) Procedura DeleteRule
- 36) Procedura GetRuleValue
- 37) Procedura GetRulesLength
- 38) Procedura GetRuleValuesChainLength

Procedury sloužící k editaci dotazů:

- 39) Procedura AddCrispValueAsk
- 40) Procedura AddFPValueAsk
- 41) Procedura AddFuzzyValueAsk
- 42) Procedura AddFuzzySetTAsk
- 43) Procedura AddFuzzySetEAsk
- 44) Procedura ModifyCrispValueAsk
- 45) Procedura ModifyFPValueAsk
- 46) Procedura ModifyFuzzyValueAsk
- 47) Procedura ModifyFuzzySetTAsk
- 48) Procedura ModifyFuzzySetEAsk
- 49) Procedura DeleteAsk
- 50) Procedura GetAskType
- 51) Procedura GetCrispValueInAsk
- 52) Procedura GetFPValueInAsk
- 53) Procedura GetFuzzyValueInAsk
- 54) Procedura GetFuzzySetTInAsk
- 55) Procedura GetFuzzySetEInAsk
- 56) Procedura GetAskLength

Procedury sloužící k vyhodnocení výsledků a správě těchto výsledků:

- 57) Procedura CheckData
- 58) Procedura Fuzzification
- 59) Procedura EvalueCuts
- 60) Procedura DeleteCuts
- 61) Procedura GetCutValue
- 62) Procedura GetCutsLength
- 63) Procedura EvalueOutput

Procedury sloužící ke správě hlavních ukazatelů, paměti a chybových proměnných:

- 64) Procedura AddPointers
- 65) Procedura SelectPointers
- 66) Procedura DeletePointers
- 67) Procedura GetPointersLength
- 68) Procedura FreeMemory
- 69) Procedura GetError

3.2 Stručný popis procedur podle uvedených bloků

Procedury sloužící k editaci vstupních a výstupních proměnných:

Tyto procedury slouží k přidávání, odebírání a prohlížení fuzzy proměnných v systému, k přidávání, modifikování, odebírání a prohlížení jim příslušících hodnot. Je možno pracovat se vstupními fuzzy proměnnými, jejichž fuzzy hodnoty jsou vyjádřeny pomocí funkce příslušnosti lichoběžníkového nebo exponenciálního tvaru. Stejně tak mohou být vyjádřeny hodnoty výstupních proměnných, pokud používáme inferenční model Mamdani. Máme však také možnost použít model Takagi-Sugeno. V tom případě budou výstupní proměnné nabývat hodnot ve tvaru regresní funkce.

Procedury sloužící k editaci báze pravidel:

Jedná se o blok procedur, umožňujících práci s bází pravidel fuzzy systému. Pravidla je možno pomocí jednotlivých procedur přidávat, modifikovat, odebírat a samozřejmě také prohlížet.

Procedury sloužící k editaci dotazů:

Procedury tohoto bloku slouží k přidávání, modifikaci, odebírání a prohlížení jednotlivých dotazů. Dotazy mohou být zadávány pomocí ostrého čísla, fuzzy čísla, některé z hodnot fuzzy proměnných, nebo pomocí obecné fuzzy množiny s funkcí příslušnosti lichoběžníkového nebo exponenciálního tvaru.

Procedury sloužící k vyhodnocení výsledků a správě těchto výsledků:

Tyto procedury patří k nejdůležitějším procedurám knihovny „fuzzylib“. Je zde umístěna procedura „Fuzzyfication“, pomocí níž dochází k fuzzyfikaci zadaných vstupů. Dále pak procedura „EvaluateCuts“, která vypočítává hodnoty všech ořezávacích koeficientů hodnot výstupních fuzzy proměnných. Procedura „EvaluateOutput“ pak provádí defuzzyfikaci výsledku a nabízí hned pět různých metod, jak tuto defuzzyfikaci provést (metoda těžiště, metoda sum, metody nejmenší z maxim, střed z maxim a největší z maxim). Za zmínku stojí též procedura „CheckData“, která slouží ke kontrole správnosti uživatelem vytvořené struktury vstupů a výstupů, aby během vyhodnocování výsledku nedocházelo ke zbytečným a těžce odhalitelným chybám.

Procedury sloužící ke správě hlavních ukazatelů, paměti a chybových proměnných:

Pomocí procedur knihovny „fuzzylib“ je možno vytvořit ne pouze jeden, ale hned několik různých autonomních fuzzy systémů. Toto je umožněno použitím ukazatelových struktur (blíže popsanych v literatuře [4]), pomocí nichž jsou do paměti ukládána veškerá data. Každý fuzzy systém je tvořen speciálně propojenými řetězci ukazatelů, jejichž počátek tvoří jediný ukazatel. Tito počáteční ukazatelé mohou být sdružení do dalšího řetězce a může tak vzniknout několik na sobě nezávislých fuzzy systémů. Toto je možné právě pomocí procedur tohoto bloku.

Funkce všech uvedených procedur knihovny „fuzzylib“ jsou včetně významu jejich vstupů, výstupů a chybových hlášení blíže popsány v literatuře [4].

4 Implementace procedur v jazyce C

Implementace jednotlivých procedur knihovny „fuzzylib“ byla provedena pomocí vyššího programovacího jazyka C. Všechny datové struktury a procedury byly vytvořeny pomocí základních příkazů jazyka C, procedur a konstant těchto knihoven: limits, float, stdlib, math. Toto jsou jedny ze základních knihoven jazyka C, odpovídající normě ANSI C, které jsou podporovány velkou většinou překladačů (pro HC12 to jsou např. překladače firem ImageCraft, Cosmic nebo Metrowerks). Pro ukládání veškerých dat byly použity pouze dynamické datové struktury (různé modifikace dynamického lineárního seznamu), což znamená, že jediným omezením množství informací vytvářeného fuzzy systému je dynamická paměť použitého procesoru. Jednotlivé procedury byly implementovány tak, aby byly co možná nejodolnější proti různým chybám, kterých se může dopustit programátor, ale i systém, na kterém program, používající tyto procedury, běží.

4.1 Program „FuzzyHC12“

Program „FuzzyHC12“ je v podstatě pouze editorem vstupních a zobrazovatelem výstupních dat fuzzy systému, tvořeného platformou HC12. Tento program tedy tvoří rozhraní mezi uživatelem a fuzzy systémem. Umožňuje přehledné a pohodlné zadávání vstupních fuzzy proměnných, výstupních fuzzy proměnných, jejich hodnot, zadávání báze pravidel a dotazů. Všechny tyto údaje jsou po zadání odeslány po sériové lince do platformy HC12, tam je z nich vyhodnocen výsledek a ten je zaslán zpět do programu „FuzzyHC12“ na PC, který přijaté výsledky zobrazí.

4.2 Program „fuzzyprg“

Program „fuzzyprg“ byl vytvořen pomocí procedur knihovny „fuzzylib“, jako ukázka vytvoření funkčního fuzzy systému pomocí těchto procedur na platformě HC12. Program „fuzzyprg“ komunikuje s programem „FuzzyHC12“ na PC pomocí sériové linky a přijímá od tohoto programu údaje, potřebné k zadání parametrů použitých procedur knihovny „fuzzylib“. Program „fuzzyprg“ vyhodnotí výsledek, a ten pak pošle zpět programu „FuzzyHC12“, který tento výsledek vhodně zobrazí.

4.3 Ukázka provedení testu

Pomocí programu „FuzzyHC12“ a „fuzzyprg“ byl sestaven fuzzy systém. Stejný fuzzy systém byl také sestaven pomocí fuzzy toolboxu programu MATLAB verze 5.3. Symboly A, B, C, D označují zlomy lichoběžníkového průběhu zleva doprava. Symboly E, F parametry exponenciálního průběhu funkce příslušnosti podle následujícího vzorce:

$$f(x) = e^{-\frac{1}{2E^2}(x-F)^2} \quad (1)$$

Tabulka 1. Tabulka hodnot vstupní proměnné Vstup 1.

Název	E	F
H1	1,2	-10
H2	0,8	-6
H3	1,6	-1
H4	1,1	5
H5	0,6	8

Tvar hodnot: exponenciální funkce (Gaussova křivka)

Tabulka 2. Tabulka hodnot vstupní proměnné Vstup 2.

Název	A	B	C	D
H1	-9,4	-9,4	-5	-2
H2	-3,6	-1	1	4
H3	1	4,2	8	9,6
H4	8,9	10	12	16
H5	12	17,3	20	20

Tvar hodnot: lichoběžník

Tabulka 3. Tabulka hodnot výstupní proměnné Výstup.

Název	E	F
H1	1,8	-10
H2	1	-5
H3	1,2	0
H4	1,68	5
H5	0,6	10

Tvar hodnot: exponenciální funkce (Gaussova křivka)

Báze pravidel:

Jestliže (Vstup1 je H1) a (Vstup2 je H1) pak (Výstup je H1) Jestliže (Vstup1 je H3) a (Vstup2 je H4) pak (Výstup je H4)
 Jestliže (Vstup1 je H1) a (Vstup2 je H2) pak (Výstup je H1) Jestliže (Vstup1 je H3) a (Vstup2 je H5) pak (Výstup je H5)
 Jestliže (Vstup1 je H1) a (Vstup2 je H3) pak (Výstup je H2) Jestliže (Vstup1 je H4) a (Vstup2 je H1) pak (Výstup je H2)
 Jestliže (Vstup1 je H1) a (Vstup2 je H4) pak (Výstup je H2) Jestliže (Vstup1 je H4) a (Vstup2 je H2) pak (Výstup je H2)
 Jestliže (Vstup1 je H1) a (Vstup2 je H5) pak (Výstup je H3) Jestliže (Vstup1 je H4) a (Vstup2 je H3) pak (Výstup je H4)
 Jestliže (Vstup1 je H2) a (Vstup2 je H1) pak (Výstup je H1) Jestliže (Vstup1 je H4) a (Vstup2 je H4) pak (Výstup je H4)
 Jestliže (Vstup1 je H2) a (Vstup2 je H2) pak (Výstup je H1) Jestliže (Vstup1 je H4) a (Vstup2 je H5) pak (Výstup je H5)
 Jestliže (Vstup1 je H2) a (Vstup2 je H3) pak (Výstup je H2) Jestliže (Vstup1 je H5) a (Vstup2 je H1) pak (Výstup je H3)
 Jestliže (Vstup1 je H2) a (Vstup2 je H4) pak (Výstup je H3) Jestliže (Vstup1 je H5) a (Vstup2 je H2) pak (Výstup je H3)
 Jestliže (Vstup1 je H2) a (Vstup2 je H5) pak (Výstup je H4) Jestliže (Vstup1 je H5) a (Vstup2 je H3) pak (Výstup je H4)
 Jestliže (Vstup1 je H3) a (Vstup2 je H1) pak (Výstup je H1) Jestliže (Vstup1 je H5) a (Vstup2 je H4) pak (Výstup je H5)
 Jestliže (Vstup1 je H3) a (Vstup2 je H2) pak (Výstup je H2) Jestliže (Vstup1 je H5) a (Vstup2 je H5) pak (Výstup je H5)
 Jestliže (Vstup1 je H3) a (Vstup2 je H3) pak (Výstup je H2)

Zadání dotazu:

Vstup1 = 7,5

Vstup2 = 1,8

Způsob defuzifikace: Střed z maxim

Při defuzzyfikaci pomocí metody „střed z maxim“ podaly programy tyto výsledky:
FuzzyHC12: 0,00 , MATLAB: -0,04

Zhodnocení:

Z výsledků je patrné, že oba programy uvádějí přibližně stejné hodnoty. To, že výsledky nejsou úplně totožné může být způsobeno například různými nepřesnostmi ve výpočtech, kdy programy pracují s čísly s pohyblivou desetinnou čárkou. Podobných testů bylo samozřejmě provedeno mnohem více a všechny s pozitivními výsledky. Dva další testy jsou uvedeny v literatuře [4].

5 Závěr

Knihovna procedur „fuzzylib“ vznikla z potřeby rozšíření možností mikroprocesorů v oblasti fuzzy modelování. Z počátku se jednalo pouze o rozšíření těchto možností u procesorů řady HC12, ale později byla knihovna zpracována tak, aby vyhovovala všem procesorům programovatelným v jazyce C podle normy ANSI C. Její implementace v jazyce C, který je jedním z nejpoužívanějších programovacích jazyků vyšší úrovně, přenositelnost a odolnost procedur vůči nejrůznějším chybám uživatele z ní dělá velmi silný nástroj pro vytváření programových aplikací s využitím přístupu fuzzy modelování.

Tento referát byl vytvořen za pomoci grantu FRVŠ 2003 číslo 1740.

Reference

1. Motorola. 68HC12 - CPU12 Reference Manual. Motorola. 1996.
2. Pokorný M. Řídící systémy se znalostní bází. Ediční středisko VŠB. Ostrava 1999. ISBN 80-7078-275-7.
3. Vysoký P. Fuzzy řízení. Ediční středisko ČVUT. Praha 1996. ISBN 80-01-01429-8.
4. Želasko P. Návrh a realizace knihovny fuzzy inferencí mikroprocesoru HC12 - Diplomová práce na VŠB-TU Ostrava. Ostrava 2002.

Annotation:

Software support for fuzzy models creating

The paper is specialized in problems of software support for creation and evaluation of fuzzy models. The subject of the paper is the library of procedures, which are realized in the C programming language. In programs, which are created in the C programming language, these procedures allow creating fuzzy models with inputs, outputs and knowledge bases, including their modifying and evaluating results. The library with procedures is complying with the ANSI C standard, which ensures its full compatibility with the C programming language compilers (ANSI C compatible) for various microcontrollers.