TECHNICKÁ UNIVERZITA V LIBERCI
Hálkova 6, 461 17 Liberec 1, CZ

Fakulta mechatroniky a mezioborových inženýrských studií

Teorie automatického řízení II.

FUZZY ŘÍZENÍ A REGULACE

Studijní materiály

Doc. Ing. Osvald Modrlák, CSc.

Srpen 2004

KATEDRA ŘÍDICÍ TECHNIKY

Katedra řídicí techniky

Obsah

Tento studijní materiál nabízí zájemcům základní informace o fuzzy regulaci, která je součástí vědní disciplíny označované jako Soft Computing, jehož základy položil americký vědec ruského původu L. A. Zadeh.

ÚVOD

V posledních letech se v praxi můžeme setkat v řízení a regulaci s přístupy a principy, které jsou založeny na poměrně nové vědní disciplíně, která je označována zkratkou SC-Soft Computing. Tato disciplína se zabývá symbiózou různých výpočetních postupů, jejichž společným jmenovatelem je odklon od klasického modelování založeného na booleovské logice, analytických modelech, ostré klasifikaci a deterministickém prohledávání.

V názvu uvedené slovo "soft", vyjadřující lexikálně "měkkost, mírnost", tady znamená "měkké" požadavky na přesnost popisovaných jevů. Mezi hlavní zástupce SC zahrnujeme **fuzzy logiku** (FL), **neuronové sítě** (NS) a **genetické algoritmy** (GA).

Fuzzy logika spočívá v rozšíření logických operátorů na **fuzzy množiny**. Teorie fuzzy množin spočívá v zavedení tzv. stupně příslušnosti prvku k množině, který může nabývat hodnot z intervalu <0,1> na rozdíl od klasické teorie množin, kdy každý prvek do množiny buď patří nebo nepatří. FL nám poskytuje jazyk s vlastní syntaxí a sématikou, který nám umožňuje bezprostřední použití kvalitativně formulovaných zkušeností a znalostí o řešeném problému.

Neuronové sítě jsou výpočetní struktury, které mají schopnost učení.

Genetické algoritmy provádějí náhodné prohledávání prostoru za pomoci imitace živé přírody. V tomto postupu probíhá modelová evoluce od vzniku jedinců přes jejich selekci a křížení až po jejich nahrazení dokonalejšími jedinci.

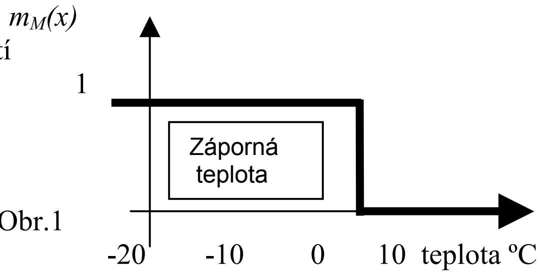
Protože cílem kurzu je pouze seznámit studenty s principy fuzzy řízení a regulace, soustředíme výklad pouze na základní charakteristiky fuzzy přístupů. Omezíme se pouze na

1) Fuzzy množiny a lingvistické proměnné
2) Operace s fuzzy množinami
3) Vyhodnocování rozhodovacích pravidel - inferenční pravidla
4) Přiřazení k výstupní fuzzy množině vhodnou ostrou hodnotu akční veličiny - defuzzyfikace
5) Strukturu fuzzy regulace

1. FUZZY MNOŽINY A LINGVISTICKÉ PROMĚNNÉ

V klasické teorii množin je možno množinu popsat několika způsoby:

a) výčtem prvků množiny $M = \{x_1, x_2, x_3, x_4\}$
b) pravidlem , kterému musí prvky vyhovovat
c) ***charakteristickou funkcí*** $m_M(x)$, pro kterou platí



$$m_M(x) \begin{cases} = 1, \text{ jestliže } x \in M \\ = 0, \text{ nepatří} \end{cases}$$

Obr.1

Příklad charakteristické funkce množiny Záporná teplota je na obr.1. Prvek *x* v klasické teorii množin do množiny bud patří, nebo nepatří, protože jeho charakteristická funkce nabývá hodnot 1 nebo 0. Hovoříme pak o **ostrých množinách** - ostrém rozlišení při rozhodování o příslušnosti.

Pokud charakteristická funkce charakterizuje stupeň, s jakým prvek do množiny patří, pak tyto množiny označujeme jako množiny **neostré - fuzzy množiny**.

V klasické teorii množin jsou definovány operace sjednocení, průnik a komplement.

Chceme-li ale využít empirických zkušeností obsluh, personálu a expertů, neobejdeme se bez zavedení a používání ***lingvistických proměnných***.

Lingvistická proměnná je taková proměnná, jejíž **hodnoty** jsou **výrazy nějakého jazyka.** **Hodnotu** lingvistické proměnné můžeme interpretovat jako fuzzy-neostré množiny.

Množina lingvistických hodnot se označuje jako množina **termů.** Termy jsou definovány na **univerzu,** které chápeme jako univerzální množinu.

Např. při regulaci teploty lázně můžeme teplotu lázně chápat jako ***lingvistickou proměnnou*** s názvem "teplota lázně". Jakou bude mít lingvistická proměnná hodnotu?

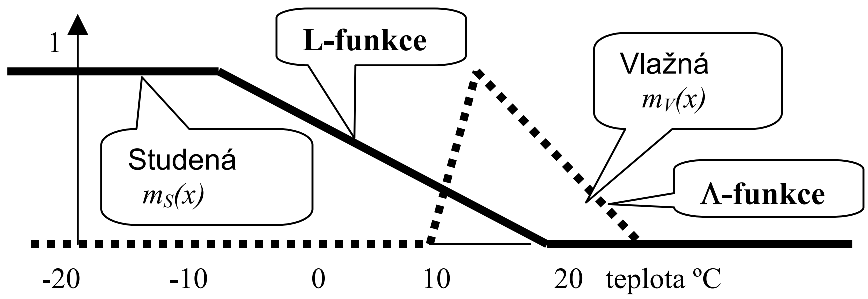
V technické praxi se v naší zemi měří teplota ve stupních Celsia. Měříme-li např. teplotu lázně, pak údaj o teplotě je ve stupních. Kvantitativní vyjádření teploty lázně v hovorovém jazyce však nemusí býti vyjádřeno jen stupni, ale běžně jsou užívány pro označení teploty výrazy jako: lázeň je LEDOVÁ, STUDENÁ, VLAŽNÁ, TEPLÁ atd.

Jako ***hodnotu lingvistické proměnné*** "teplota lázně" pak můžeme označit prvek z množiny teplot
{ ledová(L), studená(S), vlažná(V), teplá(T), horká(H) }.

Takto zavedená jazyková kvantifikace teplot (např. studená) představuje term, který označuje neostrou množinu, pro kterou je možno definovat **charakteristickou funkci** $m_S(x)$.

Charakteristická funkce $m_S(x)$ u neostrých fuzzy-množin se nazývá **funkcí příslušnosti** $m_S(x)$. Charakterizuje stupeň, s jakým daný prvek patří do příslušné množiny, a to od hodnoty 0, kdy prvek do množiny určitě nepatří, až do hodnoty 1, kdy prvek do množiny zcela určitě patří.

Jako příklad funkcí příslušnosti uvádíme neostré množiny studená a vlažná a jejich funkce příslušnosti $m_S(x)$ a $m_V(x)$ na obr.2. Každý z termů studená a vlažná je definován funkcí příslušnosti na určitém intervalu teplot (univerza) ve stupních Celsia.



Obr.2 *Funkce příslušnosti* $m_S(x)$ a $m_V(x)$

Funkci příslušnosti neostré množiny osvětlíme na následujícím příkladu, viz obr.2.

- naměříme-li teplotu x = 20 °C, pak $m_S(x) = 0$ a jistě tato naměřená teplota nepatří do termu - lingvistické hodnoty studená.

- naměříme-li teplotu x = 0 °C, pak $m_S(x) = 0,5$ což indikuje, že tato naměřená teplota patří do termu - lingvistické hodnoty studená stupněm příslušnosti 0,5.

- naměříme-li teplotu x = -10 °C, pak $m_S(x) = 1$ a je zřejmé, že tato naměřená teplota patří do termu - lingvistické hodnoty studená stupněm příslušnosti 1.

- naměříme-li teplotu x = -20 °C, pak $m_S(x) = 1$ a i tato naměřená teplota patří do termu - lingvistické hodnoty studená se stupněm příslušnosti 1.

- naměříme-li teplotu x = +15 °C, pak je $m_S(x) = 0,25$ a tato naměřená teplota patří do termu - lingvistické hodnoty studená se stupněm příslušnosti 0,25. Ale pozor, funkce příslušnosti $m_V(x) = 1$ z čehož plyne, že tato naměřená teplota patří také do množiny-termu vlažná se stupněm příslušnosti 1.

Proces přiřazování měřených hodnot vstupních veličin do fuzzy množin pomocí funkcí příslušností se označuje jako **fuzzyfikace.** Pro regulační úlohy se používají standardní funkce příslušnosti: Λ-funkce (funkce trojúhelníková), L-funkce (viz obr.2), Π-funkce (funkce lichoběžníková) viz Γ-funkce, S-funkce a Z-funkce. My se omezíme na funkce složené z lineárních úseků.

Také pro označování hodnot lingvistické proměnné se používá standardní označení. Typické označení termů - fuzzy hodnot a jejich zkratek, včetně anglického označení, je v tab.1

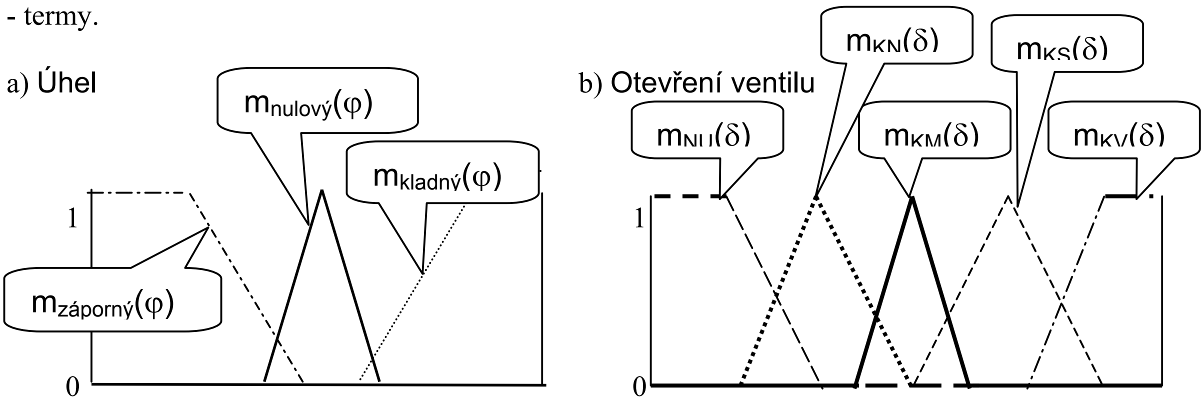
Význam	Ozn. Čes.	Ozn. Ang.
Hodnota velká záporná	ZV	NL
Hodnota střední záporná	ZS	NM
Hodnota malá záporná	ZM	NS
Hodnota záporná blízká nule	ZN	NZ
Hodnota nulová	NU	Z
Hodnota kladná blízká nule	KN	PZ
Hodnota malá kladná	KM	PS
Hodnota střední kladná	KS	PM
Hodnota velká kladná	KV	PL

Tab.1.

Příklad dalších lingvistických proměnných

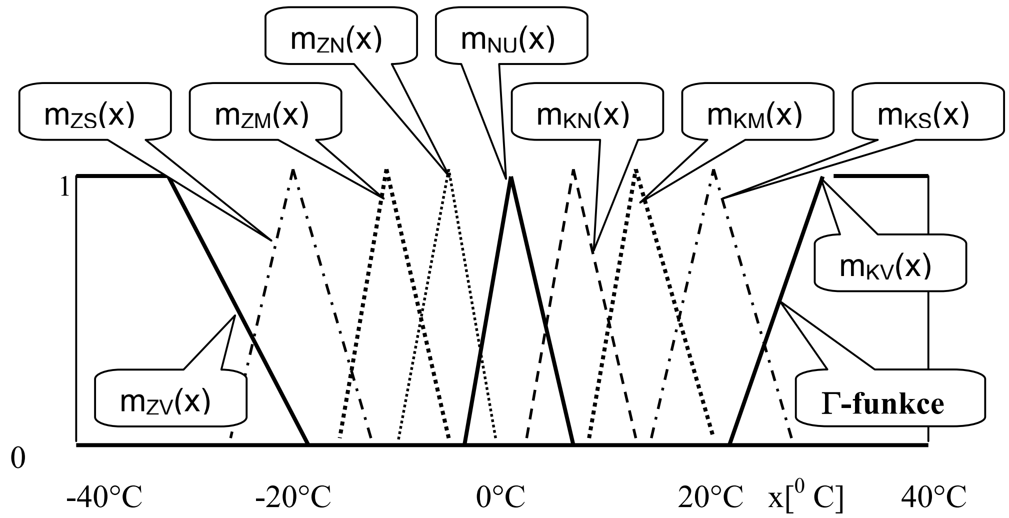
Lingvistická proměnná-označení	*lingvistické hodnoty - termy*
1. Vzdálenost	nulová, blízká, střední, veliká, obrovská
2. Úhel	záporný, nulový, kladný
3. Tepelný výkon	ZV, ZS, NU, KS, KV
4. Teplota	ZV, ZS, ZM, NU, KM, KS, KV
5. Regulační odchylka	ZV, ZS, ZM, ZN, NU, KN, KM, KS, KV
6. Otevření ventilu	NU, KN, KM, KS, KV
7. Přírůstek regulační odchylky	záporný (Z), kladný (K)

Pro lingvistické proměnné Úhel, Otevření ventilu, Regulační odchylka a Přírůstek regulační odchylky jsou na obr.3a,b,c,d zobrazeny funkce příslušnosti pro jejich lingvistické hodnoty - termy.

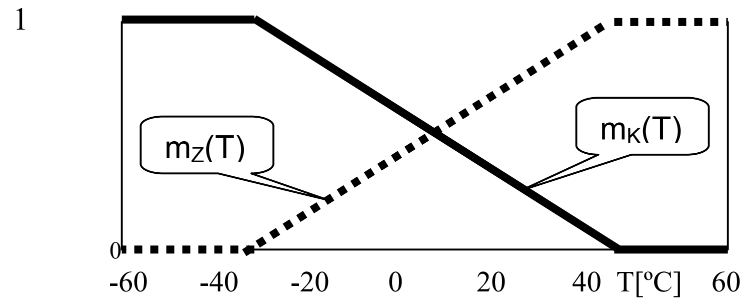


-0,5 -0,25 0 0,25 φ[rad] 0,5 0% 25% 50% 75% δ[%] 100%

c) Regulační odchylka



d) Přírůstek regulační odchylky



Obr.3 *Funkce příslušnosti pro vybrané lingvistické proměnné a jejich hodnoty - termy.*

2. OPERACE S FUZZY MNOŽINAMI

Fuzzy množiny můžeme pokládat za zobecnění klasických ostrých množin. Ostrá množina může být pokládána za zvláštní případ fuzzy množiny, jejíž funkce příslušnosti nabývá jen hodnot 0 a 1.

Operátory fuzzy logiky

Fuzzy logika spočívá pouze v rozšíření funkce logických operátorů (AND, OR, NOT) z dvouhodnotové logiky pro vícehodnotovou (fuzzy) logiku. Pro naší potřebu uvedeme pouze základní tři.

Fuzzy komplement, doplněk množiny A, C = NOT A

$$m_C(x) = 1 - m_A(x)$$

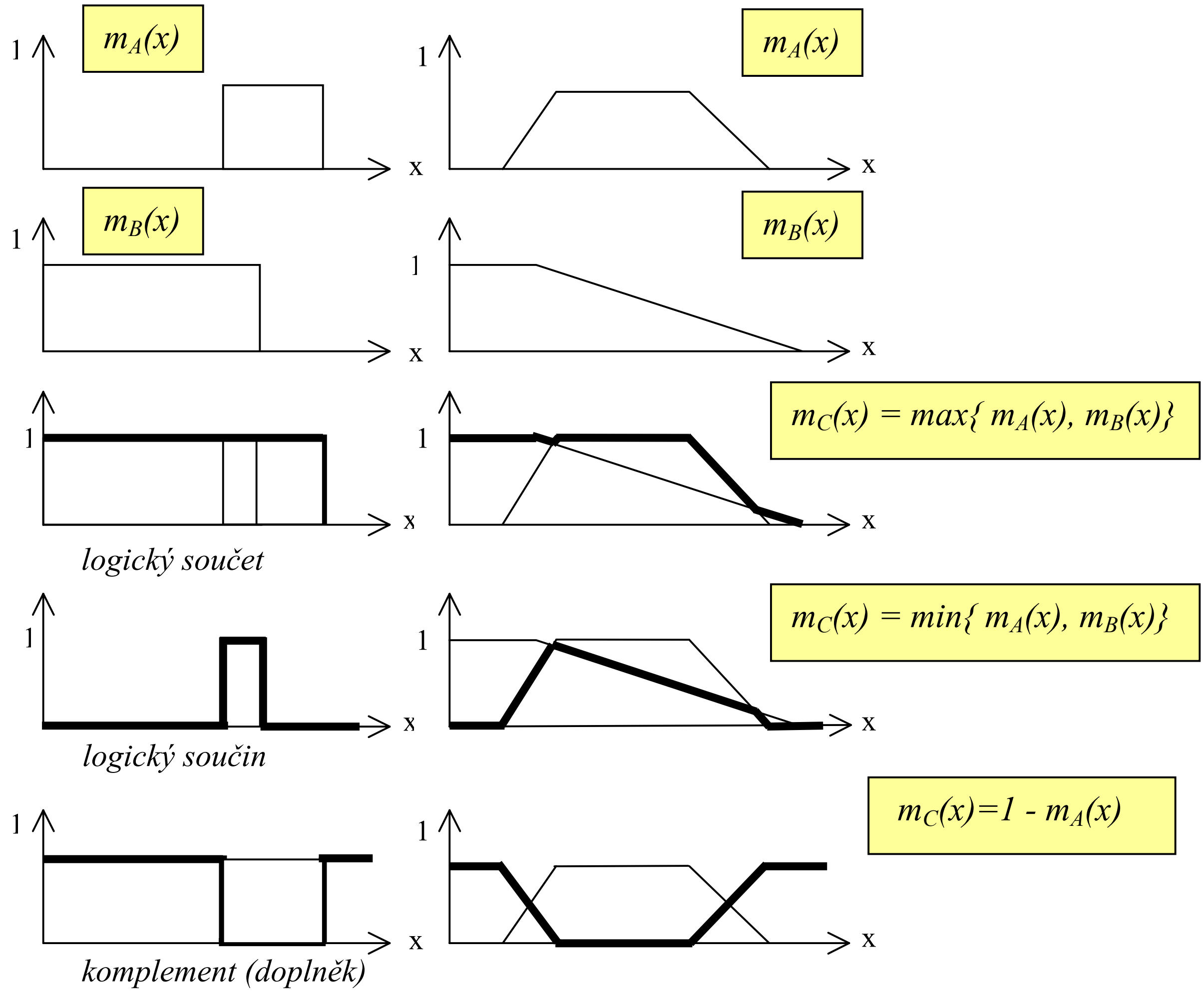
Fuzzy průnik množin (logický součin) C = A AND B

$$m_C = m_{A \wedge B}(x) = \min\{ m_A(x), m_B(x) \}$$

Fuzzy sjednocení množin (logický součet) C = A OR B

Tyto operace jsou z

$$m_C = m_{A \vee B}(x) = max\{\, m_A(x),\, m_B(x)\}$$



Obr.4 *Operace s fuzzy množinami OR, AND a komplement*

3. INFERENČNÍ PRAVIDLA

Obecně je logické řízení založeno na vyhodnocování rozhodovacích pravidel ve formě

JESTLIŽE … PAK.

Pro fuzzy řízení a regulaci je podmínka vyjádřena formou **implikace** dvou fuzzy výroků většinou jako

JESTLIŽE <fuzzy výrok> **PAK** <fuzzy výrok>,

v anglické verzi pak

IF <fuzzy výrok> **THEN** <fuzzy výrok>.

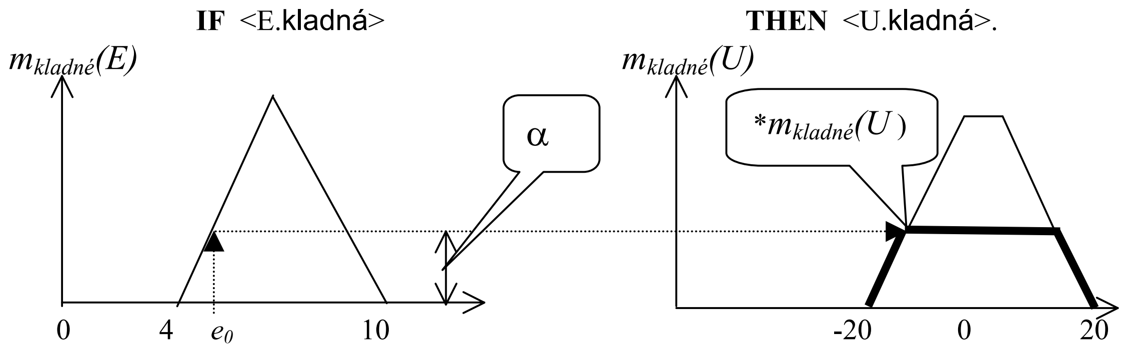
Tato podmínka je označována jako “produkční pravidlo, jestliže-pak”. První fuzzy v rokóvá množina, kterou je často složený výrok, se nazývá **ancedent**, kde jednotlivé části výroku jsou vázány logickými spojkami. Druhý fuzzy výrok je **konsekvent**.

3.1 IMPLIKACE JEDNOROZMĚRNÉ ZÁVISLOSTI

Diskutujme následující příklad. Uvažujme jednoduchý fuzzy výrok

IF <E. kladná> THEN <U. kladná>.

V rozhodovacím pravidle je v ancedentu lingvistická proměnná E (regulační odchylka), jejíž hodnota je "kladná" a má funkci příslušnosti $m_{kladné}(E)$. Konsekvent obsahuje lingvistickou proměnnou U (akční veličinu) s hodnotou "kladná", jejíž funkce příslušnosti je $m_{kladné}(U)$, viz obr.5.



Obr.5 *Implikace jednorozměrné závislosti*

Změříme-li ostrou hodnotu regulační odchylky e_0, pak můžeme v obr.5 pomocí funkce příslušnosti $m_{kladné}(E)$ odečíst stupeň příslušnosti α, s jakým změřená hodnota přísluší k množině hodnot E.kladná. Naším úkolem však je nalézt pro změřenou ostrou hodnotu odpovídající fuzzy množinu konsekventu.

Nejčastější postup jak určit tuto množinu vychází z logického předpokladu, že důsledek - konsekvent může mít maximálně stupeň příslušnosti jako má podmínka - ancedent. Stupeň příslušnosti změřené "ostré" hodnoty e_0 určuje tedy hladinu, která nám ořízne výstupní fuzzy množinu *U* konsekventu. Funkce příslušnosti konsekventu pak je $*m_{kladné}(U)$ (obrys tlustě). Tato implikace se označuje jako Mamdamiho implikace.

3.2 IMPLIKACE DVOUROZMĚRNÉ ZÁVISLOSTI S JEDNÍM PRAVIDLEM

Zobecnění tohoto principu na dvourozměrný případ

JESTLIŽE (X je kladné střední) AND (Y kladné střední) **PAK** (U je záporné střední),

vyjadřuje Mamdaniho a Larsenova implikace (bude vysvětlena později). Mamdaniho implikace definuje funkci příslušnosti konsekventu jako

$$m_{IM}(x1,x2) = min\{ m_A(x1), m_B(x2) \}$$

Minimalizací se vyjadřuje skutečnost, že důsledek (**konsekvent**) může mít maximálně stupeň příslušnosti, jako má podmínka (**ancedent**). Můžeme též hovořit o optimistickém závěru.

Nalezení výstupních množin pro dvourozměrnou závislost a jedno rozhodovací pravidlo vychází z pravidla, že pokud se pravidla překrývají, pak každé pravidlo v **ancedentu** vygeneruje svou individuální výstupní fuzzy množinu, které se také překrývají a vybíráme podle Mamdaniho minimum (vazbou v části **IF - ancedentu** je operátor **AND**). Ukažme interpretaci tohoto pravidla na následujícím příkladu:

JESTLIŽE (*x* je kladné malé) AND (*y* kladné střední) **PAK** (*u* je záporné střední),

v anglické verzi

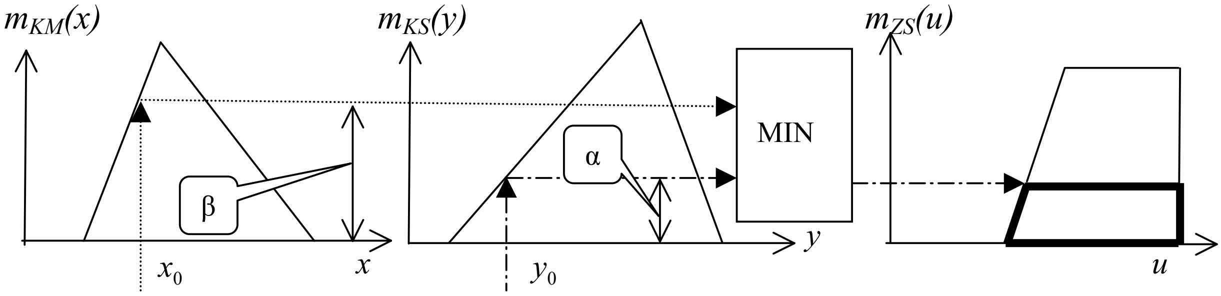
IF <x.PS> AND <y.PM> **THEN** <u.NM>

Nalezení výstupní množiny pro jedno pravidlo a dvourozměrnou závislost je na obr.6. Použitím Mamdaniho implikace obdržíme funkci příslušnosti konsekventu jako minimum z ancedentu a projekce Mamdaniho relace do osy *m*. Což znamená oříznutí funkce příslušnosti konsekventu na hladině α, která odpovídá minimu ze stupňů příslušnosti pro obě vstupní ostré hodnoty x_0 a y_0.

$$\alpha = m_{KM}(x) \wedge m_{KS}(y) = \min\{ m_{KM}(x), m_{KS}(y) \}$$

Pro funkci příslušnosti konsekventu obdržíme

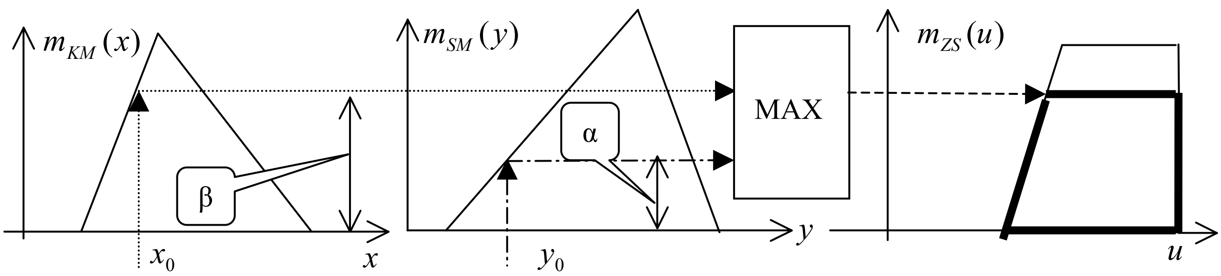
$$^{*}m_{ZS}(u) = \alpha \wedge m_{ZS}(u) = \min\{ \alpha , m_{ZS}(u) \}$$



Obr.6. *Nalezení výstupní množiny pro jedno pravidlo a dvourozměrnou závislost*

Je-li tato vazba **OR**, pak vybíráme MAXIMUM z odpovídajících funkcí příslušnosti, viz. obr.7. Oříznutí funkce příslušnosti konsekventu na hladině β, která odpovídá maximu z obou funkcí příslušnosti vstupních hodnot.

IF <x.KM> OR <x.KS> **THEN** <u.ZS>



Obr.7. *Nalezení výstupní množiny pro jedno pravidlo a dvourozměrnou závislost s operátorem OR*

3.3 IMPLIKACE DVOUROZMĚRNÉ ZÁVISLOSTI SE DVĚMA PRAVIDLY

Nalezení výstupní množiny pro dvě pravidla a dvourozměrnou závislost

IF <x.KM> AND <y.KM> **THEN** <u.KM> **ELSE**
IF <x.KS> AND <y.KM> **THEN** <u.KS>

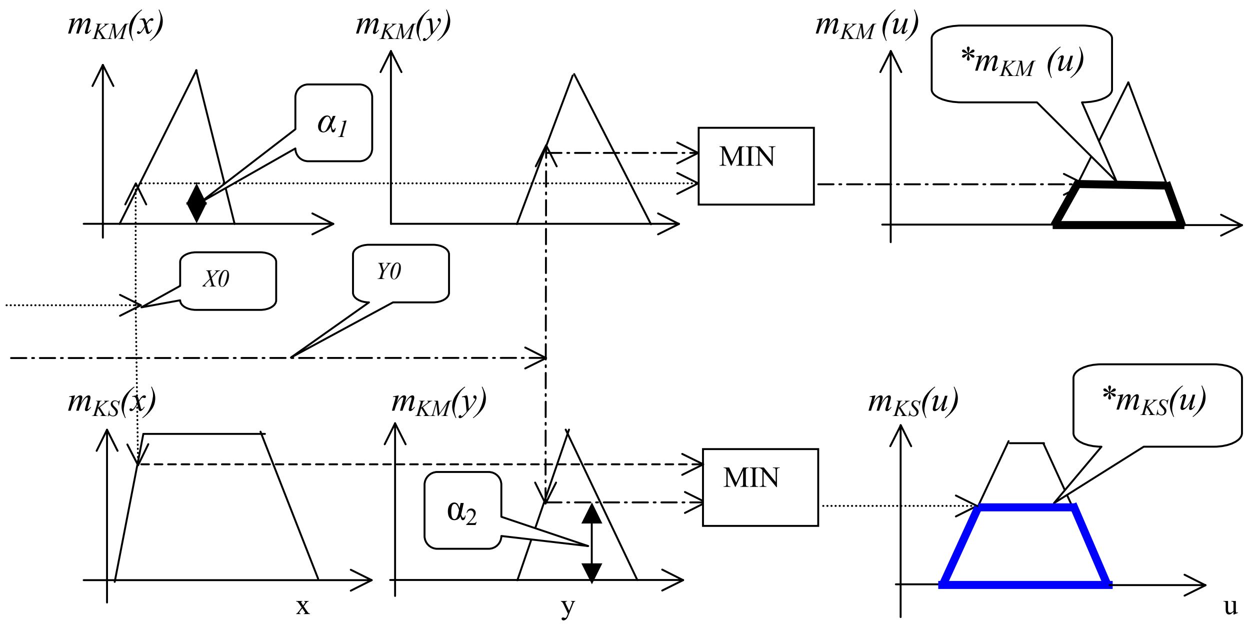
a Mamdaniho implikaci je zobrazeno na obr.8

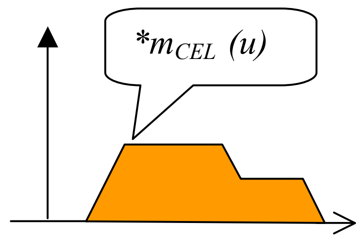
Pro dvě rozhodovací pravidla jsou popsaným postupem určeny funkce příslušnosti dvou výstupních lingvistických proměnných - termů, pro které platí

$$\alpha_1 = m_{KM}(x) \wedge m_{KM}(y) = \min \{ m_{KM}(x), m_{KM}(y) \}$$
$$\alpha_2 = m_{KS}(x) \wedge m_{KM}(y) = \min \{ m_{KS}(x), m_{KM}(y) \}$$

Pro konsekventy obou implikací dostaneme

$$*m_{KM}(u) = \alpha_1 \wedge m_{KM}(u) = \min \{ \alpha_1, m_{KM}(u) \}$$
$$*m_{KS}(u) = \alpha_2 \wedge m_{KS}(u) = \min \{ \alpha_2, m_{KS}(u) \}$$





Obr.8 *Nalezení výstupní množiny pro dvě pravidla a dvourozměrnou závislost*

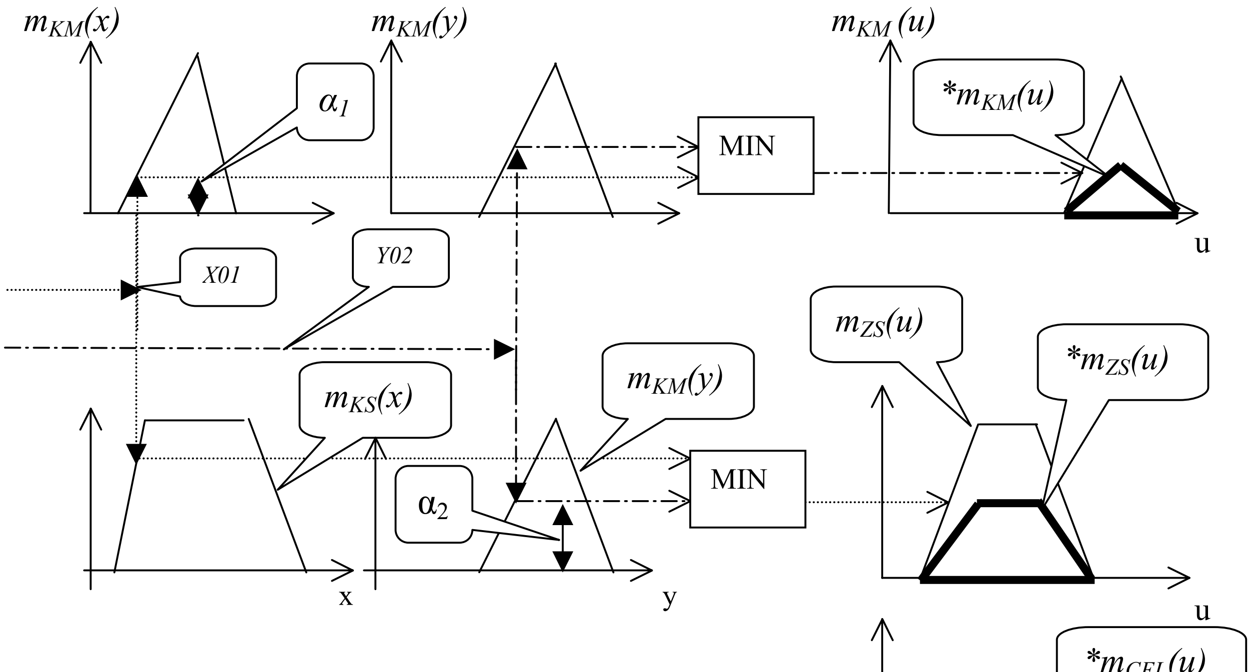
Konsekventy obou implikací $*m_{KS}(u)$ a $*m_{KM}(u)$ určují jejich dílčí podíly na velikosti akční veličiny. Intuitivně se nabízí možnost interpretovat účinek obou dílčích výstupních termů jako jejich logický součet. Pak pro výstupní fuzzy množinu obou účinků dostaneme

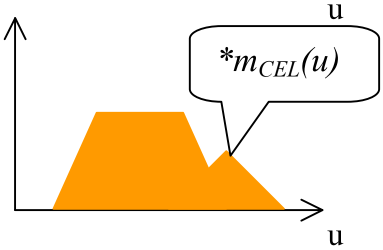
$$*m_{CEL}(u) = \max \{ \min \{ \alpha_1, m_{KM}(u)\}, \min \{ \alpha_2, m_{KS}(u) \}\}.$$

Toto pravidlo je možno rozšířit na libovolný počet rozhodovacích pravidel.

3.4 LARSENOVA IMPLIKACE

Použijeme-li **Larsenovy implikace,** pak výstupní množina pro dvě rozhodovací pravidla nebude oříznuta hladinou α1, α2, ale vynásobená těmito hladinami jak je vidět na obr.9.





Obr.9 *Inference s použitím Larsenovy implikace*

3.5 POČET PRAVIDEL

Pro dvourozměrnou funkční závislost lingvistických proměnných *X*, *Y* inferenční pravidla tvoří dvojice, které patří do množiny $A \times B$, která je dána kartézským součinem

$$P = \{(x, y) / x \in A, y \in B\}$$

Počet pravidel pro dvě fuzzy veličiny (dvourozměrná závislost: regulační odchylka e a změna reg. odchylky Δe) vysvětlíme následovně.

Regulační odchylka e má 5 lingvistických hodnot-termů (ZV, ZS, NU, KS,KV).
Změna reg. odchylky Δe má 3 lingvistické hodnoty-termy (Z, NU, K).

Protože "Regulační odchylka" je fuzzyfikována pěti termy a "Změna reg. odchylky Δe" má tři termy, je celkový počet pravidel P 5 x 3 = 15, viz obr.10. Pro počet pravidel platí

$\boxed{P = n \text{ x } m,}$ kde *m* a *n* je počet termů fuzzy množin.

Změna akční veličiny Δu má 5 lingvistických hodnot-termů (ZV, ZS, NU, KS, KV)

Δe \ e	ZV	ZS	NU	KS	KV
Z	**ZV**	**ZV**	**ZS**	**NU**	**KS**
NU	**ZV**	**ZS**	**NU**	**KS**	**KV**
K	**ZS**	**NU**	**KS**	**KV**	**KV**

Obr.10 *Báze pravidel*

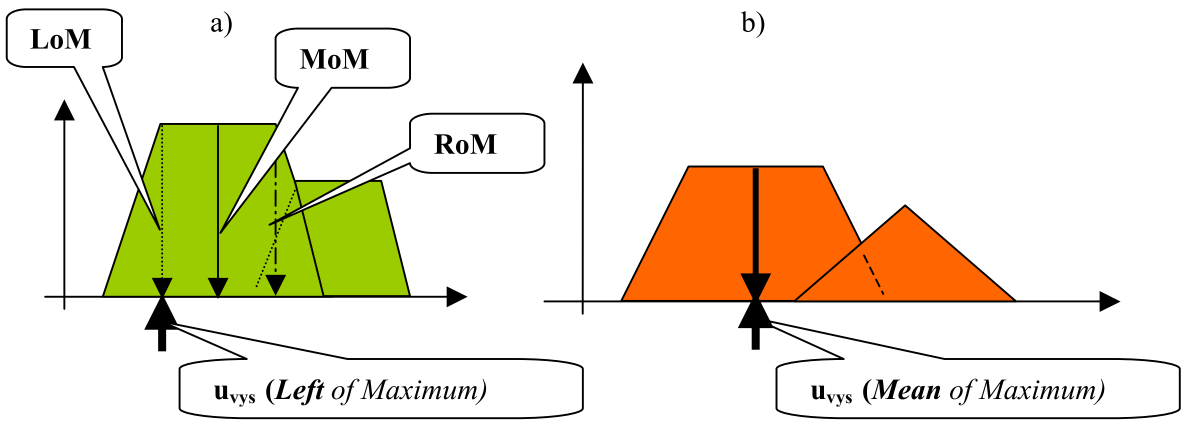
Z praktického pohledu využití fuzzy aproximací a jejich vlastností pro řízení a regulaci je třeba provést následující kroky:

1. Změřit vstupní veličiny.
2. Zobrazit změřené veličiny ve vhodném měřítku na aplikovaná univerza.
3. Převést vstupní ostrá data na fuzzy data.
4. Nalézt výstupní fuzzy množinu.
5. Přiřadit-nalézt k výstupní množině vhodnou ostrou hodnotu akční veličiny.

Krok 1 je problémem měření a nebudeme ho podrobněji diskutovat. Krok 2 je problémem normalizace a spočívá v transformaci naměřených hodnot do normovaného intervalu. Krok 3 je nazýván **fuzzyfikací** a spočívá v tom, že každé ostře naměřené hodnotě se přiřadí stupeň příslušnosti do jedné nebo více fuzzy množin. Krok 4 je podrobně rozebrán v této kapitole. Krok 5 se nazývá **defuzzyfikací** a jeho podstatou je přiřadit výstupní fuzzy množině odpovídající ostrou výstupní hodnotu. Metody používané v řízení a regulaci jsou uvedeny v následujícím odstavci.

4. DEFUZZYFIKACE

Výsledkem činnosti bloku rozhodovacích pravidel je soubor funkcí příslušnosti pro jednotlivé termy výstupních lingvistických proměnných. Funkce příslušnosti výstupní množiny je dána sjednocením oříznutých (Mamdaniho implikace) nebo zmenšených funkcí příslušnosti (Larsenova implikace), viz. obr. 8,9. Pro praktické provedení akčních zásahů je třeba přiřadit výstupním lingvistickým proměnným ostrou hodnotu akční veličiny v přípustném rozsahu. Tento proces „aproximace neostrých termů" ostrou hodnotou akční veličiny se nazývá **defuzzyfikace**. Existuje celá řada metod defuzzyfikace, které vycházejí z empirického ověření až po heuristické přístupy.



Obr.11. *Funkce příslušnosti pro jednotlivé termy výstupních lingvistických proměnných.*

Při volbě metody defuzzyfikace můžeme zvolit buď metody, které hodnotu akční veličiny určí výpočtem jako **nejlepší kompromis (*metody těžiště*)** nebo metody hledající **přijatelné řešení** (***metody nejvýznamnějšího maxima***). Přijatelné řešení:

4.1 MEAN OF MAXIMUM - METODA NEJVÝZNAMNĚJŠÍHO MAXIMA

U metod tohoto typu hledáme tzv. přijatelné řešení, které vyhovuje podmínkám v rozhodovacích pravidlech. Ze všech termů vybereme term s největší hodnotou funkce příslušnosti a nalezneme maximální hodnotu funkce příslušnosti, která pak svým umístěním v závislosti na zvolené metodě určí ostrou hodnotu výstupní veličiny.

Mezi tyto metody patří:

- ***Left of Maximum* (LoM) …**výsledkem je nejvíce vlevo položená hodnota z největší hodnoty funkce příslušnosti

- ***Mean of Maximum*** **(MoM) …**výsledkem je ve středu položená hodnota největší hodnoty funkce příslušnosti
- ***Right of Maximum*** **(RoM) …**výsledkem je nejvíce vpravo položená hodnota z největší hodnoty funkce příslušnosti

Na obr.11a je příklad určení akční veličiny u_{vys} metodou ***Left of Maximum*** a na obr.11b metodou ***Mean of Maximum***. Protože se hledá jen maximum, vyznačují se tyto metody vysokou výpočetní rychlostí. Jejich nevýhodou je, že akční veličina se může měnit nespojitě.

4.2 METODY TĚŽIŠTĚ

Metody těžiště z průběhů výstupních termů určí ostrou výstupní proměnnou jako jejich těžiště. Existují dva základní přístupy

a) **Center of Maximum** (těžiště singltonů) - funkční závislosti jednotlivých termů nahradíme jejich typickými hodnotami a hledáme jejich těžiště

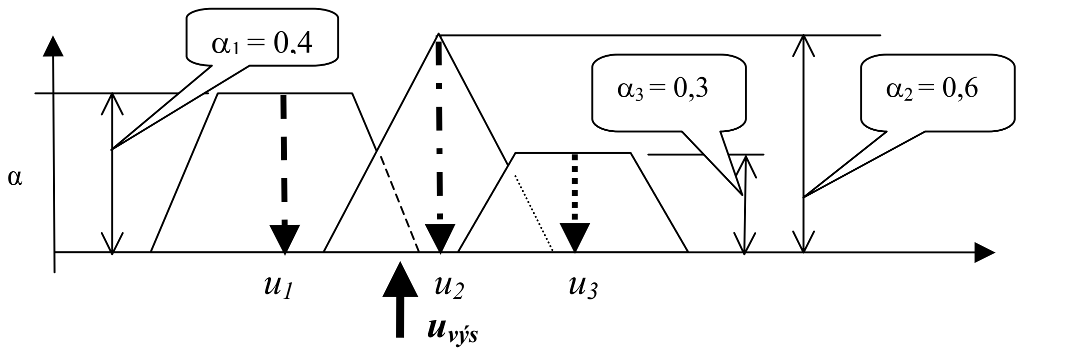
b) **Centre of Gravity** (těžiště plochy) - hledáme těžiště plochy funkce příslušnosti výstupní veličiny.

Center of Maximum

Nejdříve vysvětlíme pojem "typické hodnoty". Funkci příslušnosti můžeme aproximovat Diracovým impulsem s vahou, kterou označujeme jako "typickou hodnotu".

Poloha Diracova impulsu pro funkce příslušnosti typu Lambda funkce je ve vrcholu trojúhelníka, pro PI funkci uprostřed úseku. V některých případech je možno umístit Diracův impuls do těžiště plochy pod funkcí příslušnosti. Vlastní váha - typická hodnota může býti dána koeficientem oříznutí (násobení) α nebo β.

Metoda "Center of Maximum" nahrazuje funkční závislost každého výstupního termu jeho typickou hodnotou a **ostrou výstupní veličinu $u_{výs}$** určí jako jejich těžiště viz. obr.12.



Obr.12 *Výpočet akční veličiny metodou Center of Maximum*

$$u_{výs} = \frac{\sum_{k=1}^{r} \alpha_k * u_k}{\sum_{k=1}^{r} \alpha_k}$$

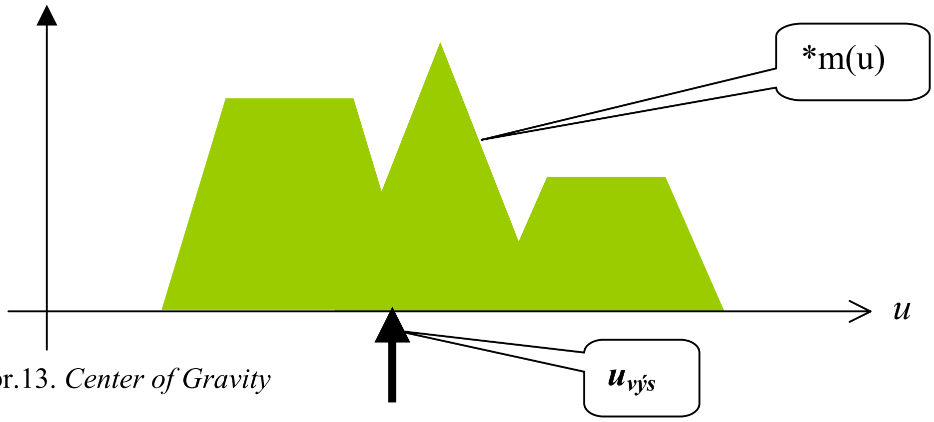
$u_{výs}$ …je výsledná hodnota výstupní veličiny

α_k …je hodnota příslušnosti k-tého termu

u_k … je souřadnice výstupní veličiny k-tého termu.

Center of Gravity

Výslednou hodnotu akční veličiny určíme jako souřadnici těžiště plochy vzniklé sjednocením dílčích ploch, které jsou určeny ohraničením funkcí výstupních termů s nenulovými hodnotami funkce příslušnosti, viz obr.13.



Obr.13. *Center of Gravity*

Výstupní hodnota akční veličiny se určí ze vztahu

$$u_{výs} = \frac{\int * m(u) \cdot u du}{\int * m(u) du}$$

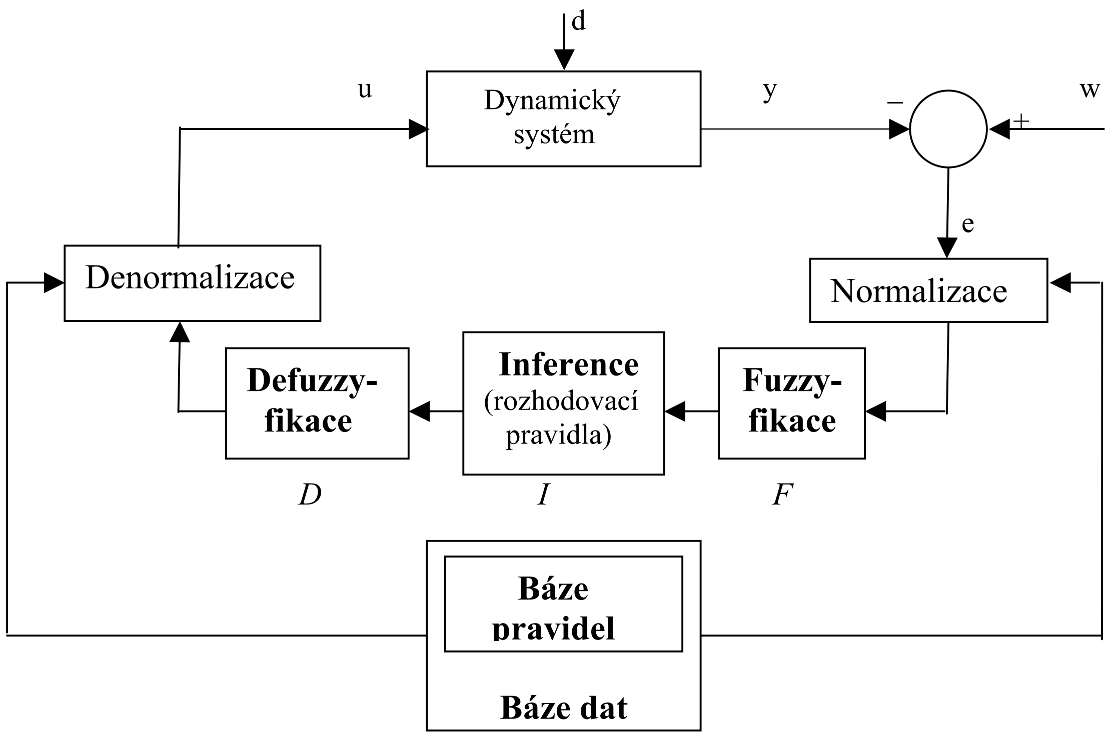
kde **m(u)* ...průběh funkce příslušnosti výsledné plochy. Pro defuzzyfikaci je možno použít ještě celou řadu metod, kterými se již nebudeme podrobněji zabývat. Je zřejmé, že každá metoda poskytuje trochu odlišné defuzzyfikované výstupy a proto jejich použití se volí podle druhu aplikace.

5. FUZZY REGULÁTORY

Charakteristickým znakem fuzzy řízení je možnost bezprostředního použití empirických znalostí člověka - operátora o řízeném procesu, které označujeme jako **bázi znalostí**. Bázi znalostí tvoří

a) informace o stacionární stavech, intervalech, ve kterých se pohybují hodnoty vstupních a výstupních veličin, jejich mezní hodnoty, atd. Rozšíříme-li tato data o funkce příslušnosti všech vstupních a výstupních fuzzy množin (jak bude vysvětleno později), pak všechny tyto informace o procesu se v bázi znalostí označují jako **báze dat**.
b) kvantitativně formulované zkušenosti včetně slovně definované strategie řízení, pomocí kterých je možno realizovat řízení, to jest generovat akční veličinu. Takto zkušeností získané strategie řízení označujeme jako **bázi pravidel**.

Struktura fuzzy regulátoru je na obr.14 a jeho ústřední člen tvoří tři základní bloky: ***fuzzyfikace F***, ***inference I*** a blok ***defuzzyfikace D***. V bloku ***fuzzyfikace*** se převádí



Obr.14 *Struktura fuzzy regulátoru*

ostrá data, která jsou naměřena nebo zadána, na fuzzy data. Bloku fuzzyfikace může předcházet blok normalizace, kde se fyzikální hodnoty naměřených či zadaných hodnot převedou na normalizovanou množinu - univerzum.

V bloku ***inference***, který tvoří ústřední část regulátoru, se realizuje inferenční mechanismus z rozhodovacích pravidel, pomocí kterého získáváme ze vstupních fuzzy množin výstupní množiny.

Blok ***defuzzyfikace*** umožňuje přiřadit výstupní fuzzy množině určitou ostrou výstupní veličinu. Za blokem ***defuzzyfikace*** může následovat blok ***denormalizace***, kde se provede denormalizace výstupní veličiny - přepočet na fyzikální výstupní veličiny.

5.1 JEDNODUCHÝ FUZZY REGULÁTOR TYPU PID

Výstup číslicového PI regulátoru v přírůstkovém tvaru, který zajišťuje nulovou regulační odchylku, je

$$u(k) = u(k-1) + \Delta u(k); \Delta u(k) = q_0 e(k) + q_1 e(k-1).$$

Výstup číslicového PD regulátoru, který ovšem nezajišťuje nulovou regulační odchylku, je ve tvaru

$$u(k) = K_p e(k) + K_D \Delta e(k).$$

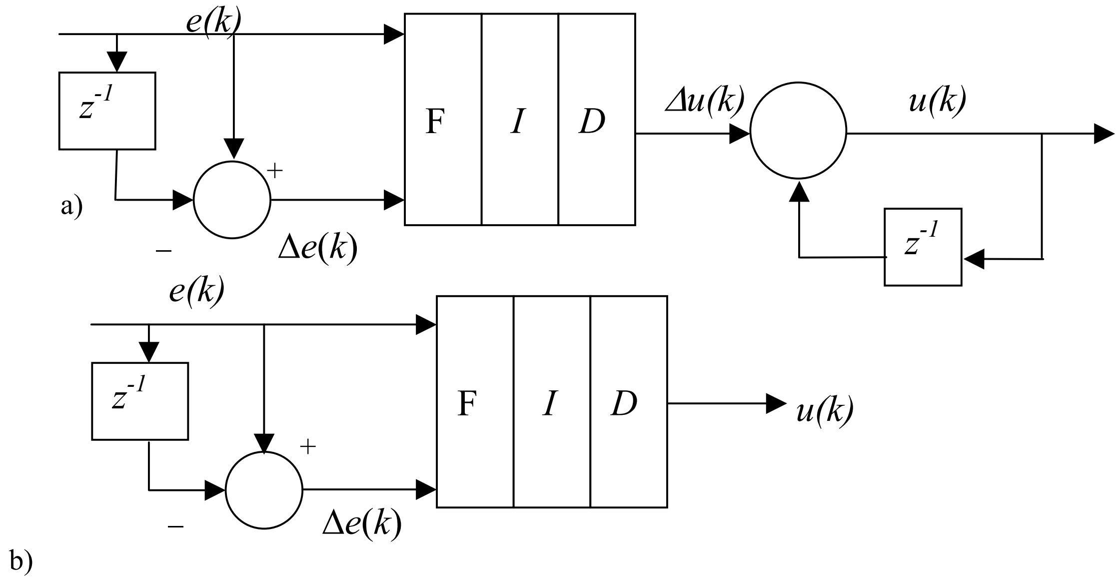
Hovoříme-li o jednoduchém fuzzy regulátoru a chceme-li ho porovnávat s PI regulátorem nebo s PD regulátorem, pak vstupem těchto regulátorů je $e(k)$ a $\Delta e(k)$. Výstup je pak nelineární funkcí, která závisí na fuzzyfikaci, inferenci a defuzzyfikaci. Takže pro fuzzy regulátor typu PI bude platit

$$\Delta u(k) = F_{PI}(e(k), \Delta e(k));\ u(k) = u(k-1) + \Delta u(k).$$

Fuzzy regulátor typu PD dostáváme jako nelineární funkci ve tvaru

$$u(k) = F_{PD}(e(k), \Delta e(k)).$$

Struktura fuzzy regulátoru typu PI a PD je na obr.15a, b.



Obr.15. *Struktura jednoduchých fuzzy regulátorů typu PI a PD*

Jednoduchý fuzzy regulátor s vlastnostmi PI-PD regulátoru vytvoříme nejjednodušším způsobem tak, že tyto dva regulátory paralelně propojíme viz [2,3].

5.2 TVORBA BÁZE PRAVIDEL

Bázi pravidel je možno vytvořit buď a) na základě empirických znalostí obsluhy nebo
b) na základě obecně platných metapravidel.

Praxe ukázala, že pro jednoduchý fuzzy regulátor typu PI, PD je možno odvodit bázi pravidel pomocí tří základních metapravidel, která uvedeme:

MP1: Jestliže regulační odchylka $e(k)$ a její změna $\Delta e(k)$ je nulová nebo blízká nule, pak by měl být přírůstek akční veličiny $\Delta u(k)$ – akční zásah nulový nebo blízký nule.

MP2: Jestliže regulační odchylka $e(k)$ klesá k nule nebo se blíží nule s dostačující rychlostí, pak je vhodné také neměnit akční veličinu.

MP3: Jestliže se regulační odchylka $e(k)$ nekoriguje sama, potom je třeba akční veličinu změnit a akční zásah $\Delta u(k)$ bude nenulový. Jeho velikost a znaménko závisí na znaménku a velikosti regulační odchylky $e(k)$ a její změny $\Delta e(k)$.

Podle těchto metapravidel byla pro jednoduchý fuzzy regulátor typu PI (odstraňuje trvalou regulační odchylku), lingvistické proměnné a jejich hodnoty / termy sestavena báze pravidel, která je uvedena v tab. č.2.

Regulační odchylka	e	{ Z, N, K}
Změna regulační odchylky	Δe	{ Z, N, K}
Akční zásah	Δu	{ Z, N, K}

e \ Δe	*Z*	*N*	*K*
Z	Z	Z	N
N	Z	N	K
K	N	K	K

Tab.2

1\.

V bázi pravidel je možno rozlišit pět skupin pravidel.

Skupina

Tato skupina pravidel se použije tehdy, jestliže regulační odchylka $e(k)$ a její změna $\Delta e(k)$ je nulová nebo blízká nule. Znamená to, že regulovaná soustava je v ustáleném stavu nebo v jeho blízkosti.

Akční veličina se nemá měnit, čili změna akční veličiny je nulová nebo blízká nule.

2\.

Skupina

Pro aplikaci pravidel této skupiny platí, že regulační odchylka $e(k)$ je záporná (velká nebo střední) a její změna $\Delta e(k)$ je kladná nebo blízká nule. Důsledkem toho je, že regulační odchylka $e(k)$ se **zmenšuje nebo se nemění**.

Akční zásah má zrychlit nebo zpomalit přibližování k ustálené hodnotě.

3\.

Skupina

Pro tuto skupinu platí, že regulační odchylka $e(k)$ je kladná (blízká nule, střední, veliká). Změna $\Delta e(k)$ je kladná velká nebo střední, což znamená, že regulovaná veličina **se bude vzdalovat** od žádané hodnoty - ustáleného stavu.

Kladnou změnou akční veličiny $\Delta u(k)$ je třeba zajistit přibližování k ustálenému stavu.

4\.

Skupina

Pro aplikaci pravidel této skupiny je charakteristické, že regulační odchylka $e(k)$ je kladná (velká nebo střední) a její změna $\Delta e(k)$ je záporná nebo nulová. To znamená, že regulační odchylka $e(k)$ *se* **zmenšuje nebo se nemění**. (Porovnej se skupinou 2)

Akční zásah má zrychlit nebo zpomalit přibližování k ustálené hodnotě.

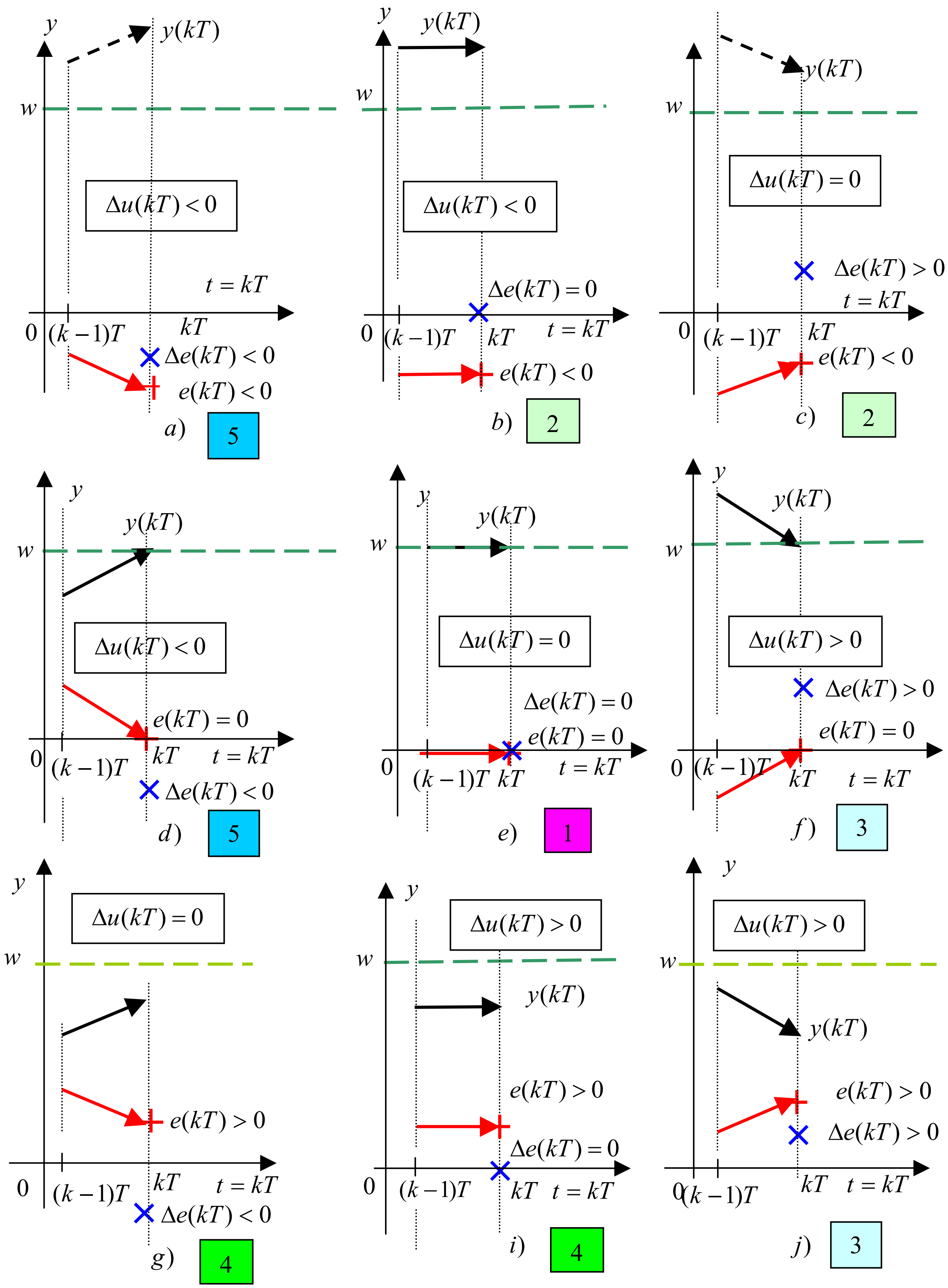
5\.

Skupina

Pro tuto skupinu platí, že regulační odchylka $e(k)$ je záporná (blízká nule, střední, veliká). Změna $\Delta e(k)$ je **záporná velká nebo střední**. To znamená, že regulovaná veličina se bude **vzdalovat** od žádané hodnoty - ustáleného stavu. (Porovnej se skupinou 3)

Zápornou změnou akční veličiny $\Delta u(k)$ je třeba zajistit přibližování k ustálenému stavu.

Grafické zobrazení výstupní veličiny $y(kT)$, žádané hodnoty w, regulační odchylky $e(kT)$, změny regulační odchylky $\Delta e(kT)$ a změny akční veličiny $\Delta u(kT)$ pro předpokládané varianty regulované veličiny v krocích $k, k-1$ je na obr.15-1.



Obr.15-1. *Grafické zobrazení regulační odchylky* $e(kT)$ *a změny regulační odchylky* $\Delta e(kT)$ *a změny akční veličiny* $\Delta u(kT)$ *pro konstrukci báze pravidel v Tab. č.2*

Velká většina jednoduchých fuzzy regulátorů má bázi pravidel založenou na použití uvedených pravidel. Báze pravidel lze snadno modifikovat pro jiný počet termů regulační odchylky a její změny, viz. příklad Tab.3.

NB	NB	NB	NB	NM	NS	Z
NB	NB	NB	NM	NS	Z	PS
NB	NB	NM	NS	Z	PS	PM
NB	NM	NS	Z	PS	PM	PB
NM	NS	Z	PS	PM	PB	PB
NS	Z	PS	PM	PB	PB	PB
Z	PS	PM	PB	PB	PB	PB

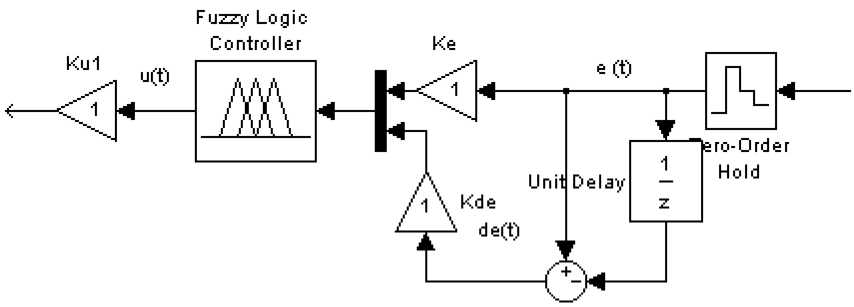
Tab.3 *Regulační odchylka* *e* *{ NB, NM, NS, Z, PS, PM, PB }*
Změna regulační odchylky *Δe* *{ NB, NM, NS, Z, PS, PM, PB }*
Akční zásah *Δu* *{ NB, NM, NS, Z, PS, PM, PB }*

Průběh regulačního pochodu ovlivňuje kromě báze pravidel také zvolené tvary funkcí příslušností a zvolená metoda defuzzyfikace. V případě, že průběhy regulačních pochodů nevyhovují zcela našim požadavkům, je třeba hledat nová rozhodovací pravidla, použít jiných metod defuzzifikace a upravit vhodně funkce příslušnosti.

5.3 SEŘÍZENÍ JEDNODUCHÉHO FUZZY REGULÁTORU TYPU PID

Odezvy regulačních obvodů s fuzzy regulátorem závisí na bázi rozhodovacích pravidel a na bázi dat. Součástí projektu fuzzy regulátoru je získání či vytvoření báze rozhodovacích pravidel, zadání funkcí příslušnosti pro jednotlivé vstupní a výstupní proměnné, včetně volby metod fuzzyfikace a defuzzifikace. Vlastní implementace těchto bázických znalostí pro daný řídící systém nebo produkt se realizuje softwarově. V předmětech "Teorie řízení" využíváme softwarové podpory MATLABu, speciálně pak "Fuzzy toolboxu", viz kap.6. Je zřejmé, že na dynamiku regulačních pochodů má vliv celá řada parametrů, jejichž účinky na dynamiku soustavy lze jen těžko odhadnout. Z těchto důvodů je nastavení všech hledaných parametrů pokládáno jako velmi obtížné.

Omezíme se proto na nastavování fuzzy regulátoru pomocí měřítek univerza, v anglické literatuře se hovoří o "Tuning via scaling universes". Princip metody je velmi jednoduchý a spočívá ve vážení - násobení konstantou vstupní i výstupní proměnné fuzzy regulátoru, viz obr. 16. Pomocí vah



Obr.16. *Struktura fuzzy regulátorů s vahami pro seřízení regulátoru.*

na vstupu K_e a K_{de} měníme vlastně měřítka univerza na vstupu a pomocí zesílení K_u měníme měřítka na výstupu - akční veličiny. Pokud nedosáhneme požadovaných průběhů regulačních pochodů, je nutno použít jiných postupů, což přesahuje rámec našeho předmětu.

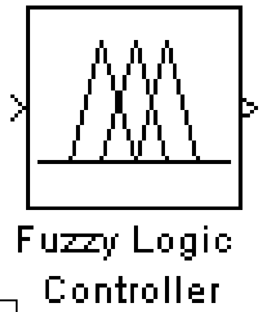
6. FUZZY LOGIC TOOLBOX

Návrh fuzzy regulátoru v MATLABu se provádí pomocí programu Fuzzy Logic Toolbox. Simulační výpočty včetně reálných měření se provádí v SIMULINKU.

6.1 IMPLEMENTACE FUZZY LOGIKY A ŘÍZENÍ V SIMULINKU

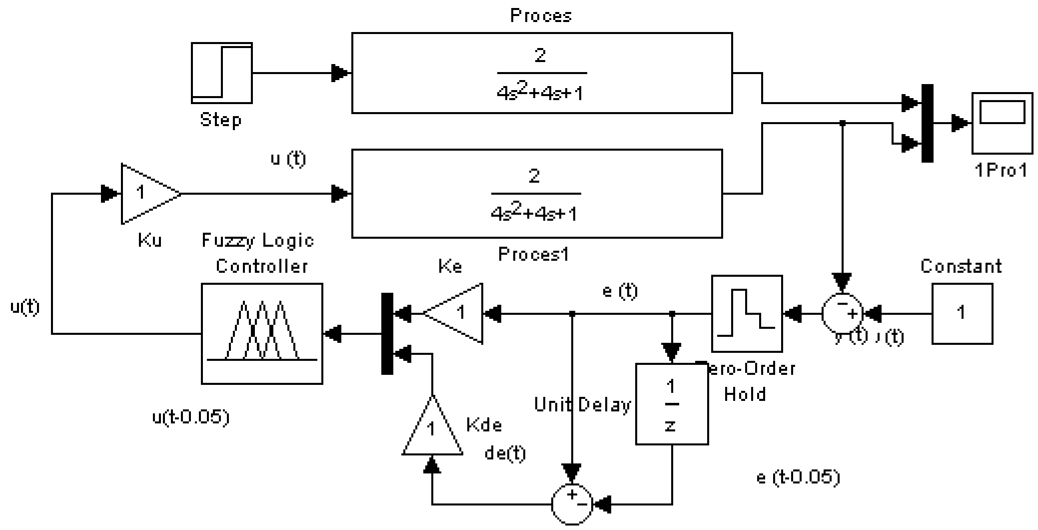
Pro regulaci v reálném čase, měření a simulace se používá SIMULINK, který při aktivaci Fuzzy Logic Toolboxu obsahuje v menu Fuzzy Logic Toolboxu fuzzy regulátory:

a) Fuzzy logic controller viz obr.17
b) Fuzzy logic controller with Ruleviewer



Obr.17

Fuzzy regulátor se v SIMULINKu propojuje běžným způsobem, viz obr.18. Základní strukturu fuzzy regulátoru, kterou tvoří bloky fuzzyfikace, defuzzyfikace a inference, zastupuje v SIMULINKu blok "Fuzzy logic controller". Struktura uzavřeného regulačního obvodu se spojitou soustavou 2. řádu a s fuzzy regulátorem s diskretizovanými vstupy je na obr.18. Regulační obvod zajišťuje změny žádané hodnoty a pro srovnání je možno regulovanou soustavu nezávisle budit skokovými změnami. Ve zpětné vazbě je možno měnit zesílení na vstupu do fuzzy regulátoru pomocí zesílení K_e a K_{de} a dále ještě na výstupu z fuzzy regulátoru. Tato zesílení slouží k ladění - seřizování fuzzy



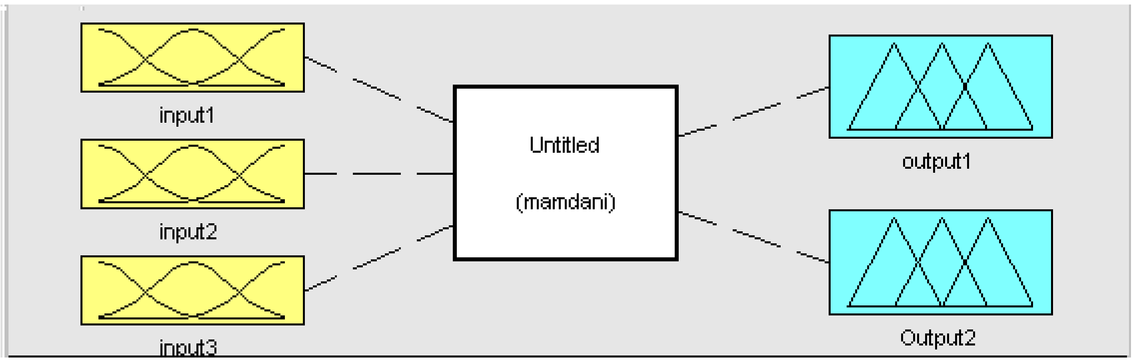
Obr.18 *Základní struktura zpětnovazebního obvodu s fuzzy regulátorem*

regulátoru. Podle požadavků na průběh regulační odchylky je možno do obvodu zapojit integrátor (sumátor) nebo změnit či upravit strukturu členů zapojených do zpětné vazby.

Při klasické implementaci regulátoru typu PID se nejdříve volí jeho struktura (P, PI, PD, PID) a pak se provádí seřízení jeho parametrů. Regulátor typu PID pouze vyhodnocuje a zpracovává informace o regulační odchylce *e(t)*. Výsledkem tohoto procesu je výstupní veličina z regulátoru *u(t)*. Fuzzy regulátor využívá další možné informace o procesu (báze pravidel a dat) včetně informací a zkušeností obsluhy. Obsahem návrhu fuzzy regulátoru je pak získání těchto informací a jejich využití v rámci návrhu fuzzy regulátoru.

6.2 NÁVRH STRUKTURY A VLASTNOSTÍ FUZZY REGULÁTORU

Vlastní návrhářskou práci fuzzy regulátoru (FR) je pak možno provádět pomocí interaktivního grafického prostředí Graphical User Interface (GUI) nebo pomocí příkazové řádky Command Line (CL). Projektování vyžaduje definovat vstupní a výstupní proměnné, jejich rozsahy, funkce příslušnosti a jejich parametry, zadávání inferenčních a rozhodovacích pravidel, nastavení metod fuzzyfikace a defuzzyfikace. Pro tyto požadavky můžeme vyjádřit strukturu FR blokově dle obr.19. Tuto strukturu, jako **metodickou projektovou pomůcku**, v souladu s Fuzzy Logic Toolboxem budeme značit jako **Fuzzy Inference System FIS** - Inferenční systém fuzzy. Uživatelské grafické prostředí GUI obsahuje nástroje pro vytvoření, editaci a zobrazování fuzzy inferenčního systému (FIS). Fuzzy inferenční systém (FIS) zahrnuje všechny procesy spojené s volbou vstupů regulátoru včetně jejich parametrizace až po určení výstupů z regulátoru s použitím fuzzy logiky - obr.19.



Obr.19 *Fuzzy Inference System FIS - Inferenční systém fuzzy*

FIS tvoří 3 editory: **FIS Editor** (editor inferenčního systému fuzzy regulátoru), viz obr.20, **Membership Function Editor** (editor funkcí příslušnosti), viz obr.21 a **Rule Editor** (editor pravidel EP), viz obr.23 a dvě zobrazování - Rule a Surface Viewer (Grafické zobrazování procesu inference, viz obr.24 a plochy ohraničující prostor generovaných akčních zásahů, viz obr.25).

FIS Editor se při vytváření nového fuzzy inferenčního systému aktivizuje příkazem

```
Fuzzy
```

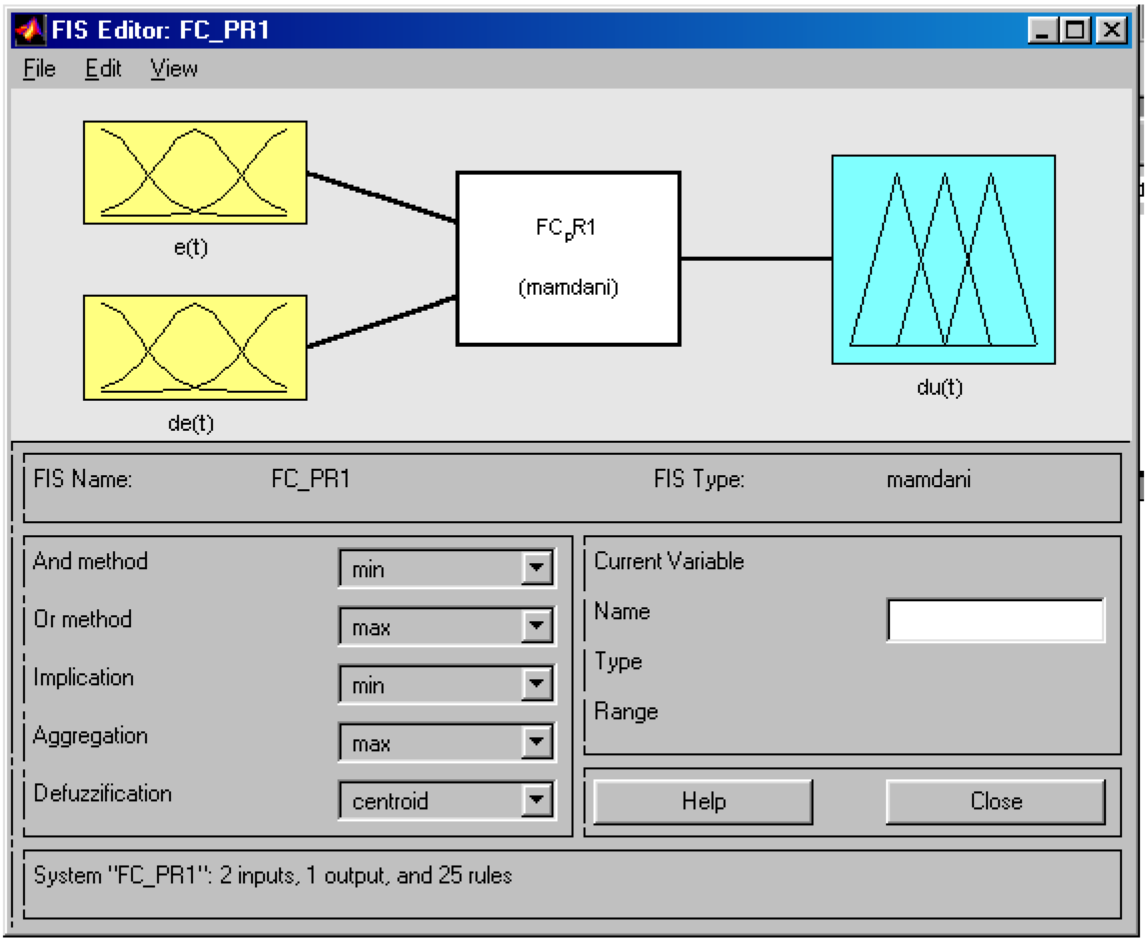
Aktivace již existujícího fuzzy inferenčního systému se aktivizuje příkazem

```
fuzzy jméno.fis
```

Tímto příkazem se vyvolá již existující fuzzy inferenční systém definovaným jménem souboru.

6.2.1 FIS Editor

Hlavní menu viz obr.20 obsahuje roletová menu File, Edit, View, která umožňují ukládání a volání souborů a editaci fuzzy systému pomocí nástojů GUI. V nabídce Edit je možno přidáním nebo ubráním určit počet vstupů a výstupů. V grafickém okně jsou v principu zobrazovány třemi ikonami - bloky: vstupní proměnné (rychlost, zrychlení), typ



Obr.20 *FIS Editor*

inference FIS (motorek1), výstupní proměnné (I-motoru). Jsou-li spojovací linky mezi bloky vyznačeny čárkovanou čarou, pak nejsou jednotlivé bloky parametrizovány nebo parametrizace není správně ukončena a tento blok není možné zapojit a spustit.

Dvojím kliknutím na vybranou vstupní proměnnou reprezentovanou obrázkem je možno přejít do Membership Function Editor (editoru funkcí příslušnosti).

Dvojím kliknutím na typ inference reprezentovanou obrázkem je možno přejít do Rule Editoru (editoru rozhodovacích pravidel).

Dvojím kliknutím na vybranou výstupní proměnnou reprezentovanou obrázkem je možno přejít do Membership Function Editor (editoru funkcí příslušnosti EF).

V levé části okna je možno zadávat příslušné parametry metodám AND, OR a Implikaci a parametry agregační a defuzzyfikační metodě. V pravé části je možno editovat jména vstupních a výstupních proměnných. Zobrazeny jsou též rozsahy proměnných a typ.

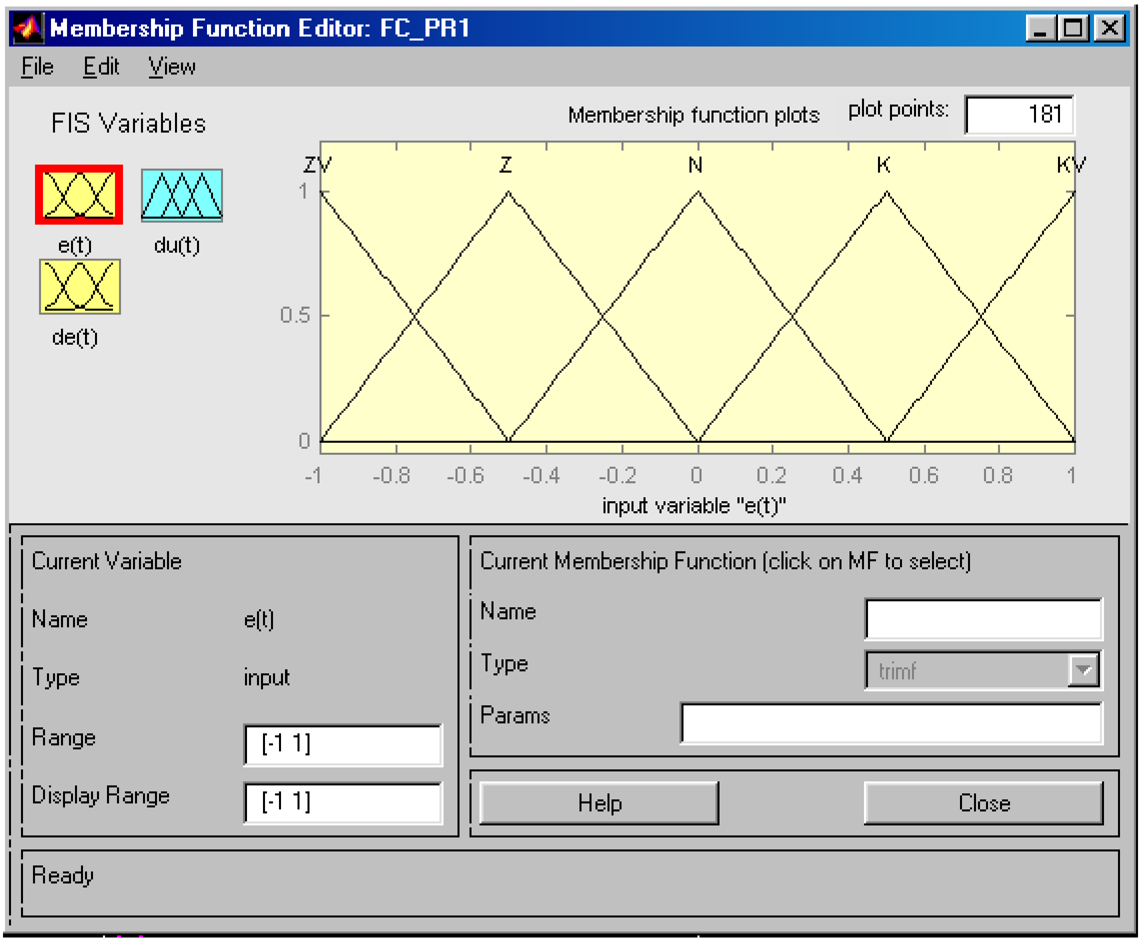
Příklad 1: Nastavte parametry FIS : 2 vstupní proměnné *e(t)*, *de(t)*, výstupní proměnná u(t), FIS uložit pod jménem FC_PR1

Postup: 1) kliknutím rozbalíme Edit a klikneme na Add input - vytvoří se dva bloky vstupů
2) klikneme na ikonku-blok input1 a v bílém poli u hesla Name přepíšeme input1 na e(t). Return
3) klikneme na ikonku-blok input2 a v bílém poli u hesla Name přepíšeme input2 na de(t). Return

4) klikneme na ikonku-blok output1 a v bílém poli u hesla Name přepíšeme output1 na u(t). Return
5) Z menu File vybereme Save to disk as FC_PR1
6) Zadáme jméno souboru PŘI_1 a klikneme na OK.

6.2.2 Membership Function Editor (MF editor)

Spustí se buď ve FIS editoru dvojím kliknutím na ikonu výstupu nebo vstupu nebo přes roletové okno **Membership Function Editor** .

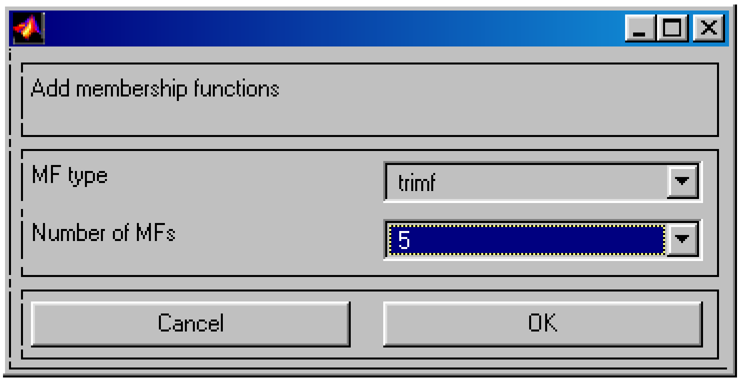


Obr.21 *Membership Function Editor*

Vybereme-li na obr.21 vstupní veličinu (*e(t)*, *de(t)*) kliknutím na její ikonu v levém rohu, vybraná ikona po obvodu zčervená a v grafickém okně "Memership function plots" se zobrazí všechny její funkce příslušnosti s nastavenými parametry a se jmény jejich proměnných - termů.

V okně "Current Variable" je uvedeno jméno, typ, rozsah a rozsah displeje označených hesly: Name, Typ, Range, Display Range. V tomto okně můžeme zadávat potřebné rozsahy. Je vhodné nejdříve u zvolené proměnné nastavit její rozsahy. Pak kliknutím v rolovacím manu Edit na příkaz Add FMs se zobrazí okno "Add membership functions" dle obr.22.

V rolovacím menu hesla MF type se volí typ funkce příslušnosti vstupní nebo výstupní proměnné (lingvistické) z množiny (trimf, trapmf, gbellmf, gausmf, gaus2mf, pimf, dsigmf, psigmf - "trimf" je trojúhelníková funkce příslušnosti).



Obr.22 *Okno" Add membership functions"*

V rolovacím menu hesla Number of MFs se volí počet termů - hodnot (lingvistické) vstupní nebo výstupní proměnné.

Klikneme-li na jednu z funkcí příslušnosti, změní tato barvu na červenou a je možno ji změnit včetně jména, tvaru a číselných parametrů. Tyto změny provedeme v dílčím okně "Current Membership Function" .

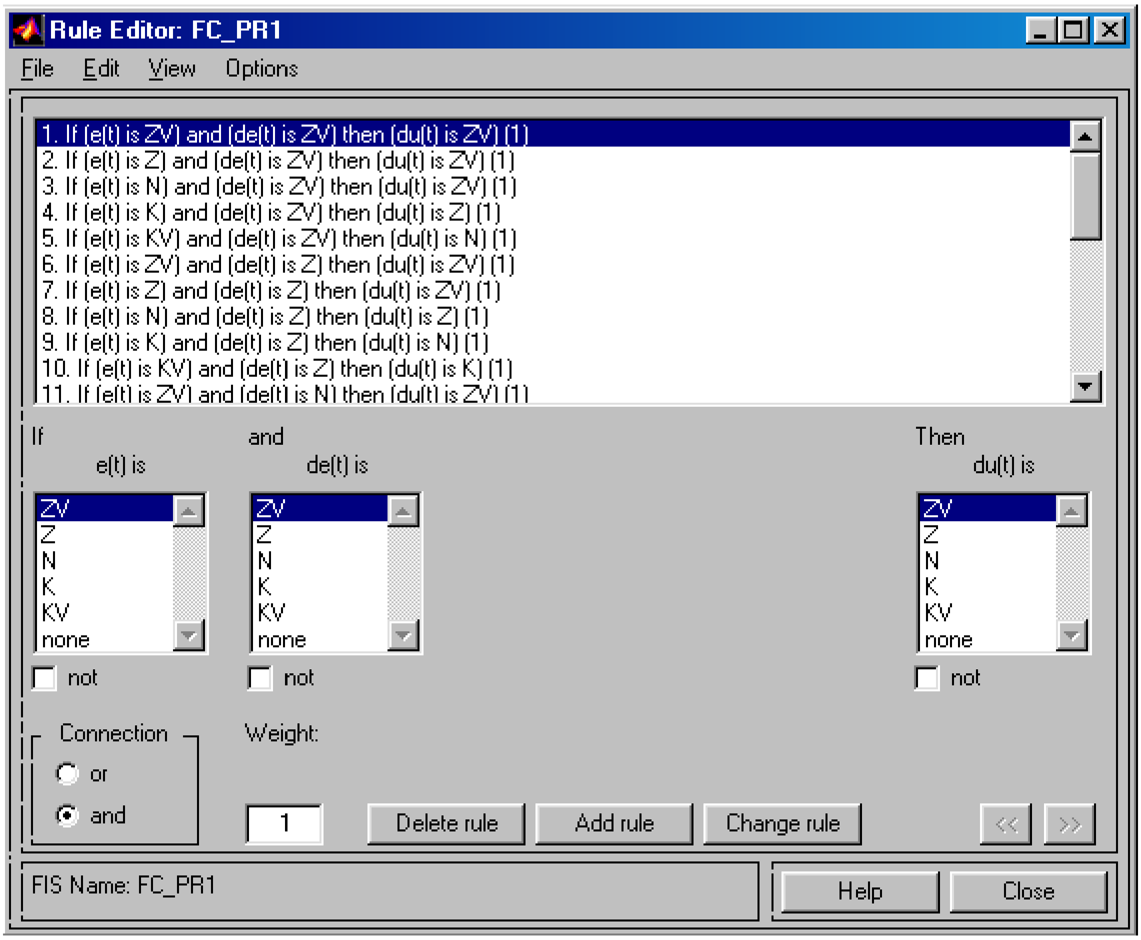
Jméno, typ a parametry vybrané funkce příslušnosti jsou zobrazeny v polích hesly: Name, Typ, Params, na kterých je také možno přepsáním jména a novým nastavením typu a parametrů funkce příslušnosti provést požadované změny.

6.2.3 Rule Editor (Editor pravidel EP)

Spustí se ve FIS editoru přes roletové okno **View-Edit Rules**. Obsahuje editační a zobrazovací pole, viz obr.23. V tomto poli je možno pravidla přímo editovat ručně nebo použít tlačítek

Delete rule : maže pravidlo
Add rule : přidává pravidlo
Change rule : mění pravidlo

Vlastní pravidlo je možno sestavit pomocí roletových menu vstupních a výstupních lingvistických proměnných (*e(t)*, *de(t)* a *u(t)*) , včetně zadávání vah. Rule Editor nabízí roletová menu pro vstupní i výstupní proměnné, kde každou položku tvoří jméno lingvistické proměnné - termu. Termy lze spojovat operátory and nebo or. Jednotlivé termy mohou vystupovat v rozhodovacích pravidlech i v negaci, což provedeme kliknutím na příslušný operátor not.



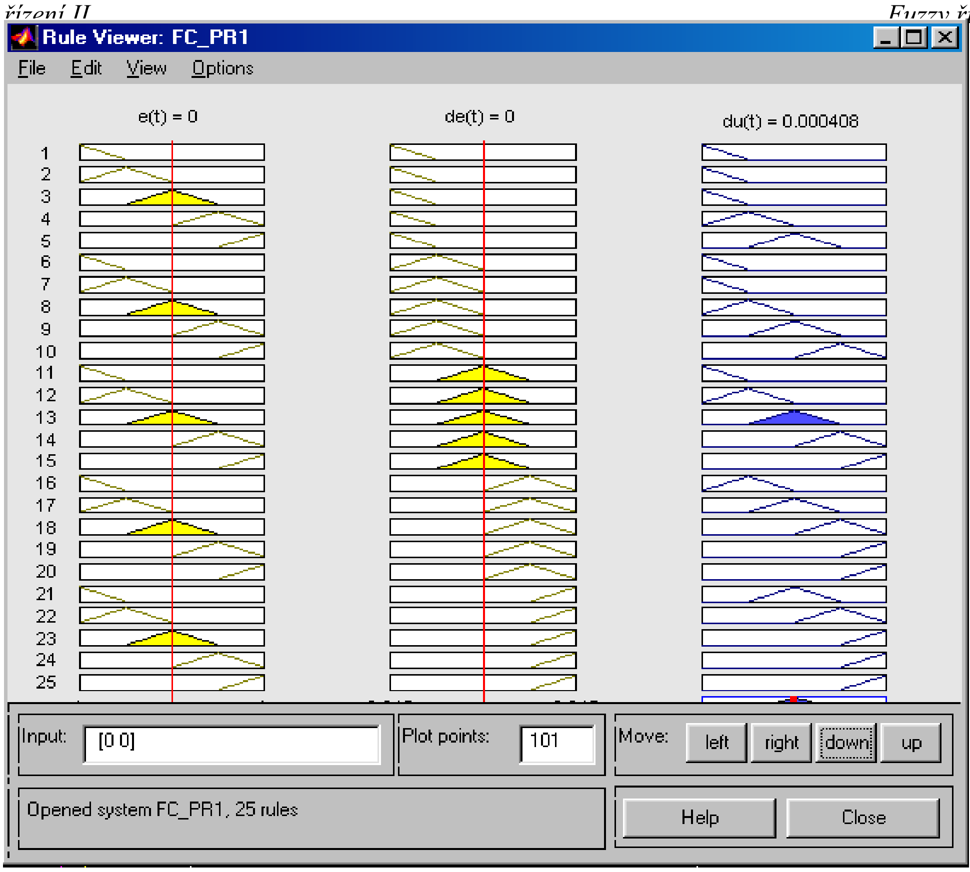
Obr.23. *Rules Editor*

6.2.4 Rule Viewer, Surface Viewer

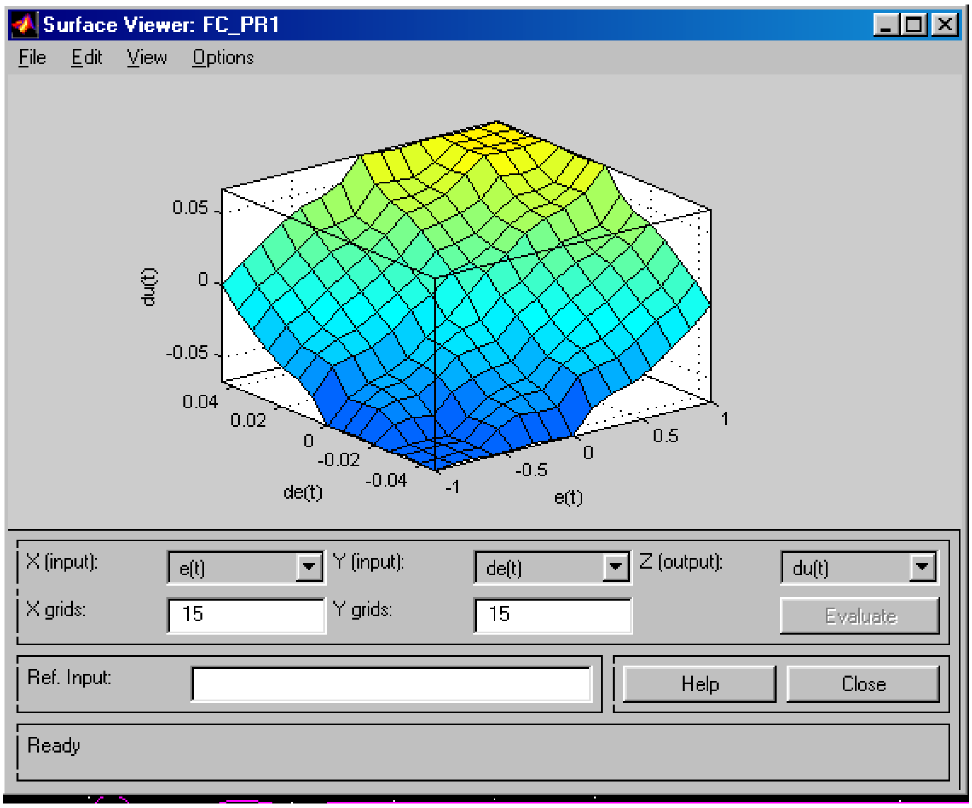
Rule Viewer (grafické zobrazování procesu inference), viz obr.24 se aktivizuje pomocí roletových menu **View** výběrem **Rule Viewer**. Obsahuje jak všechna pravidla, tak i tvary funkcí příslušnosti vstupů a výstupů a jejich inference.

Surface Viewer (grafické zobrazování procesu inference), obr.25, se aktivizuje pomocí roletových menu **View** výběrem **Surface Viewer**. Zobrazuje prostor hodnot výstupní veličiny v závislosti na vstupních proměnných. Pro fuzzy regulaci je uvažována zpravidla regulační odchylka a její derivace.

Poznámka:	Kontrolou formální správnosti nastavení FIS systému pomocí FIS Editoru je, že po skončení práce jsou vstupní bloky, blok inferencí a výstupní bloky propojeny tučnými (nepřerušovanými) čarami. V dílčím informačním okně "Systém" jsou uvedeny základní informace: jméno, počet vstupů a výstupů, počet pravidel.



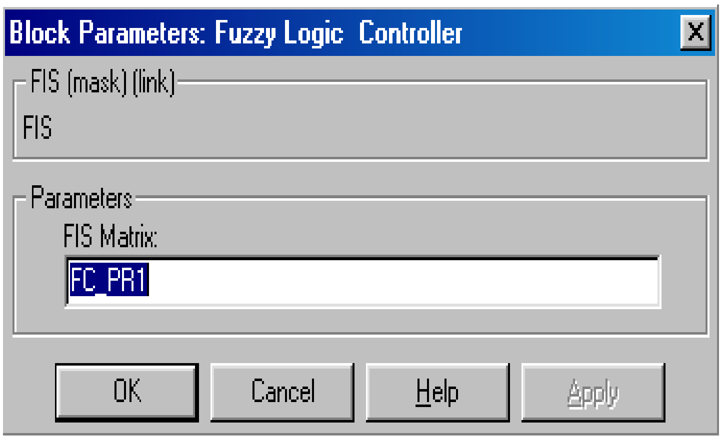
Obr.24 *Rule Viewer*



Obr.25 *Surface Viewer*

6.3 FIS MATICE

Aby bylo možno spustit simulaci v SIMULINKu, je třeba Fuzzy Inference System uložit pod jeho jménem do pracovního prostředí MATLABu (workspace). Protože v MATLABu je základem maticový popis i způsob ukládání informací, jsou informace uložené v Fuzzy Inference System FIS uloženy jako matice. Tato matice je označována jako **FIS Matrix (FIS Matice)**. Uložení do pracovního prostředí se provede na hlavní liště v rolovacím menu File příkazem Save to workspace. Dvojím kliknutím na ikonu fuzzy regulátoru v SIMULINKu se zobrazí okno "Block Parameters: Fuzzy Logic Controller" a je možno zadat aktuální jméno FIS - Matice, viz obr.26. Znamená to tedy, že pro každou volbu báze dat a rozhodovacích pravidel se vytvoří odpovídající FIS Matice. SIMULINK umožňuje vybranému fuzzy regulátoru ve schématu přiřadit zvolenou FIS matici, čímž jsou vlastnosti regulátoru definovány. V daném programovém schématu lze pak již pouze měnit váhy na vstupu a výstupu.



Obr.26 *Okno "Block Parameters: Fuzzy Logic Controller"*

Závěrem této kapitoly je třeba zdůraznit, že předložený návod k používání FUZZY LOGIC TOOLBOXu se omezuje na vysvětlení základních kroků v grafickém prostředí GUI. Nezabývá se vůbec návrhem fuzzy regulátoru pomocí příkazové řádky Command Line a vyžaduje proto pro hlubší pochopení další studium [4,5,6].

Závěr

Předložený studijní materiál má umožnit nejen získat základní informace o fuzzy přístupech a fuzzy regulaci, ale především má připravit studenty pro aplikaci Fuzzy Toolboxu a tím pro získání praktických zkušeností při aplikaci těchto metod v laboratořích katedry.

LITERATURA

[1] Passino K.M., Yurkovich S.: **Fuzzy control.** Addison Wesley Longman, Inc., Menlo Park, California, 1998, ISBN 0-201-18074-X

[2] Vysoký, P.: Fuzzy řízení. Skripta, ČVUT Praha, 1997.

[3] Pivoňka, P.: **Analysis and Design of Fuzzy Controller**. In: Fuzzy Control. Theory and Praxis, Physica-Verlag, 2000, ISBN 3 - 7908-1327-3.

[4] Gulley, N., Jang,J.S.: **Fuzzy Logic Toolbox**. For Use with MATLAB.The Math Works, Inc.1995

[5] **The Student Edition of MATLAB.**Version 4, User's Guide.The Math Works, Inc. 1995, Prentice Hall, Englewood Cliffs. ISBN 0-13-184979-4

[6] **SIMULINK** Dynamic System Simulation for MATLAB.Using Simulink, Version 2 The Math Works, Inc.1997