00bMCS 00bMCS 00bMCS 00bMCS 00bMCS 00larsen 00bLCSA 00bMCS 00bLCSA 00bPID1 00bPID1 00bPID2 00bPID1 00bFCD 00bFLwEA 00bPID1 00bFcfsdfc 00bFE 00bFcfsdfc 00bIFC 00bFcfsdfc 00bFoFC 00bFLMaFC 00bFcfsdfc 00bFLMaFC 00bFLMaFC 00bFLMaFC 00wiki 00bFLMaFC 00bFLMaFC 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFCD 00bFCD 00flt 00flt 00bFCD 00bFcfsdfc 00bFCH 00bFCH 00Luo1995 00bFcfsdfc 00bFLMaFC 00bFCH 00bFCH 00bFLMaFC 00daiflc 00bFCH 00bFCH 00bFCD 00bFCH 00bFcfsdfc 00bFCH 00daiflc 00daiflc 00bFcfsdfc 00bFLMaFC 00bFcfsdfc 00bFLMaFC 00bFCD 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFCH 00bFcfsdfc 00bFCH 00bIFC 00bFCD 00bIFC 00imps 00bFoFC 00bFoFC 00bFoFC 00bFoFC 00bFoFC 00bFoFC 00bFoFC 00bFoFC 00bFoFC 00bFoFC 00bFcfsdfc 00bFCD 00bFcfsdfc 00bFcfsdfc 00bFLMaFC 00minimp 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc 00bFcfsdfc

author1hash=BKfamily=Bart, familyi=B., given=Kosko, giveni=K.,

author1hash=CDSWfamily=Clarence, familyi=C., suffix=W., suffixi=W., given=De Silva, giveni=D. S.,

author2hash=CHHfamily=Constantine, familyi=C., suffix=H., suffixi=H., given=Houpis, giveni=H., hash=SSNfamily=Stuart, familyi=S., suffix=N., suffixi=N., given=Sheldon, giveni=S.,

author2hash=DHSfamily=D., familyi=D., suffix=S., suffixi=S., given=Hooda, giveni=H., hash=VRfamily=Vivek, familyi=V., given=Raich, giveni=R.,

labeltitlesourcetitle

author1hash=JJfamily=Jan, familyi=J., given=Jantzen, giveni=J.,

author3hash=KMfamily=Kai, familyi=K., given=Michels, giveni=M., hash=FKfamily=Fra familyi=F., given=Klawonn, giveni=K., hash=RKfamily=Rudolf, familyi=R., given=Kruse, giveni=K.,

author2hash=KsJfamily=Karl, familyi=K., suffix=J., suffixi=J., given=Åström, giveni=s., hash=THfamily=Tore, familyi=T., given=Hägglund, giveni=H.,

author1hash=LRfamily=Leonid, familyi=L., given=Reznik, giveni=R.,

author1hash=LPfamily=Li, familyi=L., given=P., giveni=P.,

author2hash=LJfamily=Luo, familyi=L., given=Jia, giveni=J., hash=LEfamily=Lan, familyi=L., given=Edward, giveni=E.,

author1hash=MLfamily=Martin, familyi=M., given=Larsen, giveni=L.,

author2hash=OAAfamily=Osama, familyi=O., suffix=A., suffixi=A., given=Abihana, giveni=A., hash=OGRfamily=Oscar, familyi=O., suffix=R., suffixi=R., given=Gonzalez, giveni=G.,

author2hash=RDCfamily=Richard, familyi=R., suffix=C., suffixi=C., given=Dorf, giveni=D., hash=RBHfamily=Robert, familyi=R., suffix=H., suffixi=H., given=Bishop, giveni=B.,

author2hash=SAfamily=Sakly, familyi=S., given=A., giveni=A., hash=BMfamily=Benrejel
familyi=B., given=M., giveni=M.,
author3hash=SSWfamily=Su, familyi=S., suffix=Whan, suffixi=W., given=Sung,
giveni=S., hash=JLfamily=Jietae, familyi=J., given=Lee, giveni=L., hash=IBLfamily=In-
Beum, familyi=I.-B., given=Lee, giveni=L.,
author1hash=TRJfamily=Timothy, familyi=T., suffix=J., suffixi=J., given=Ross,
giveni=R.,
author1hash=Wfamily=Wikipedia contributors, familyi=W.,
author2hash=ZKfamily=Zdenko, familyi=Z., given=Kovacic, giveni=K., hash=SBfamily=S
familyi=S., given=Bogdan, giveni=B.,

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Konstrukce fuzzy regulátorů

2025

Martin Hrabal

Vedoucí práce:
doc. RNDr. Michal Krupka, Ph.D.

Studijní program: Informatika, prezenční forma

Bibliografické údaje

Autor:	Martin Hrabal
Název práce:	Konstrukce fuzzy regulátorů
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2025
Studijní program:	Informatika, prezenční forma
Vedoucí práce:	doc. RNDr. Michal Krupka, Ph.D.
Počet stran:	47
Přílohy:	1 CD/DVD
Jazyk práce:	český

Bibliographic info

Author:	Martin Hrabal
Title:	Fuzzy controller design
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2025
Study program:	Computer Science, full-time form
Supervisor:	doc. RNDr. Michal Krupka, Ph.D.
Page count:	47
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

Tato bakalářská práce je zaměřena na proces vývoje fuzzy regulátorů a řešením praktických problémů, které mohou při jejich vývoji vznikat. Součástí práce je i stručný popis řídících systémů, matematická definice fuzzy logiky a sestrojení jednoduchého fuzzy regulátoru pro stabilizaci obráceného kyvadla. Fuzzy systém pro stabilizaci obráceného kyvadla je vytvořen v programu MATLAB, za pomocí rozšíření Simulink a Fuzzy logic toolbox.

Synopsis

This bachelor's thesis is focused on the process of designing fuzzy controllers and dealing with practical problems, that may appear in the process. This thesis also contains short description of control systems, mathematical definition of fuzzy logic and design of a simple fuzzy controller for stabilization of an inverted pendulum. Fuzzy system for the inverted pendulum stabilization is created in the program MATLAB, using Simulink and Fuzzy logic toolbox extensions.

Klíčová slova: Fuzzy regulátor, fuzzy logika, fuzzy systém, řídící systémy

Keywords: Fuzzy controller, fuzzy logic, fuzzy system, control systems

Tímto bych chtěl poděkovat doc. RNDr. Michalu Krupkovi, Ph.D za poskytnutí mnoha vhodných rad a odborné vedení mé práce.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

1 Úvod **8**

2 Teorie řízení **8**
2.1 Řídící systém . . . 8
2.1.1 Systémy s otevřenou smyčkou . . . 9
2.1.2 Systémy s uzavřenou smyčkou . . . 9
2.2 Lineární a nelineární systémy . . . 10
2.2.0.1 Additivita . . . 10
2.2.0.2 Homogenita . . . 10
2.3 PID regulátor . . . 11
2.3.1 PID regulátory v praxi . . . 12

3 Fuzzy regulátor **13**
3.1 Fuzzy logika . . . 13
3.1.1 Fuzzy množiny . . . 13
3.2 Operace na fuzzy množinách . . . 14
3.2.1 T-normy a S-normy . . . 15
3.2.2 α-řez fuzzy množiny . . . 16
3.2.3 Konvexní fuzzy množina . . . 16
3.3 Fuzzy relace . . . 17
3.3.1 Skládání fuzzy relací . . . 17
3.3.2 Relace podobnosti . . . 18
3.4 Rozšiřující obal . . . 19
3.5 Škálový faktor . . . 20
3.6 Normalizace . . . 20
3.7 Lingvistické proměnné . . . 21
3.7.1 Funkce fuzzy regulátoru . . . 21
3.7.1.1 Využití fuzzy regulátoru . . . 22
3.7.1.2 Mamdani a Takagi-Sugeno-Kang fuzzy systémy . . . 22
3.8 Kdy použít fuzzy regulaci? . . . 23

4 Návrh fuzzy regulátoru **23**
4.1 Průzkum informací o systému . . . 24
4.2 Rozdělení univerza na fuzzy podmnožiny . . . 24
4.2.1 Výběr funkce příslušnosti . . . 25
4.2.1.1 Trojúhelníková funkce příslušnosti . . . 25
4.2.1.2 Lichoběžníková funkce příslušnosti . . . 26
4.2.1.3 Gaussova funkce příslušnosti . . . 26
4.3 Sestrojení báze pravidel . . . 27
4.3.1 Jak vytvořit pravidla . . . 27
4.3.1.1 Kompletnost . . . 28
4.3.1.2 Konzistence . . . 29
4.3.1.3 Spojitost . . . 31

4.4 Fuzzifikace . . . 32
4.4.1 Operátor fuzzy implikace . . . 32
4.4.1.1 Funkce implikace . . . 32
4.4.1.2 T-normy . . . 34
4.4.2 Operátor agregace . . . 35
4.5 Inferenční metoda . . . 35
4.5.1 Normalizace a denormalizace . . . 36
4.6 Defuzzifikace . . . 36
4.6.1 Center of gravity . . . 36
4.6.2 Mean of maxima . . . 38
4.7 Takagi-Sugeno-Kang systém . . . 39
4.8 Příklad konstrukce fuzzy regulátoru . . . 40
4.9 Popis systému . . . 40
4.10 Volba vstupních proměnných . . . 41
4.11 Volba funkcí příslušnosti . . . 42
4.12 Tvorba pravidel . . . 42
4.13 Volba inferenční metody . . . 43
4.14 Volba defuzzifikační metody . . . 43

Závěr **44**

5 Obsah přiloženého datového média **45**

Seznam obrázků

1 Systém s otevřenou smyčkou . . . 9
2 Systém s uzavřenou smyčkou . . . 10
3 Robot . . . 11
4 Fuzzy množina A s funkcí příslušnosti μ_A . . . 14
5 α-řez fuzzy množiny . . . 16
6 Nekonvexní fuzzy množina (vlevo), konvexní fuzzy množina (vpravo) 17
7 Rozšiřující obal pro jednoprvkovou množinu x . . . 19
8 Termy lingvistických proměnných (studená, teplá a horká) [5] . . 21
9 Obecné schéma fuzzy regulátoru . . . 22
10 Rozdělení univerza na tři fuzzy podmnožiny . . . 25
11 Trojúhelníková funkce příslušnosti . . . 26
12 Lichoběžníková funkce příslušnosti . . . 26
13 Gaussova funkce příslušnosti . . . 27
14 Znázornění inference . . . 28
15 Znázornění inference nekompletní báze pravidel . . . 29
16 Znázornění inference konzistentní báze pravidel . . . 30
17 Znázornění inference nekonzistentní báze pravidel . . . 30
18 Znázornění inference nespojité báze pravidel . . . 31
19 Proces fuzzifikace . . . 32
20 Grafické znázornění defuzzifikace pomocí metody COG . . . 37
21 Výstupní fuzzy množina složená ze dvou disjunktních fuzzy množin 37
22 Interpolace funkce $f(x) = x$ pomocí Mamdaniho ovladače . . . 39
23 Vozík s kyvadlem . . . 41
24 Vstupní proměnná pro úhel náklonu kyvadla . . . 42
25 Vstupní proměnná pro rychlost kyvadla . . . 42
26 Vstupní proměnná pro polohu vozíku . . . 42
27 Vstupní proměnná pro rychlost vozíku . . . 43

Seznam tabulek

1 Fuzzy relace R: x je finanční fond s riskovým faktorem y [7] . . . 17
2 Fuzzy relace R': x je finanční fond s možností zisku/ztráty z [7] . 18
3 Fuzzy relace R': y je riskový faktor s možností zisku/ztráty z [7] . 18
4 Pravdivostní tabulka fuzzy implikace $High \Rightarrow Low$ [6] . . . 34

1 Úvod

Téma fuzzy logiky jsem si zvolil z části kvůli mému zájmu o robotiku a řídící systémy. Téma mi přišlo zajímavé a myslím si, že se dá uplatnit ve spoustě odvětví průmyslu. Zadáním bakalářské práce bylo popsat kroky pro navrhnutí kvalitního fuzzy regulátoru. V práci jsem se věnoval i obecnému popisu řídících systémů a PID regulátorům, pro které je fuzzy regulátor alternativou.

V práci jsem popsal funkci Mamdaniho fuzzy regulátoru a kroky jeho procesu. Poskytnul jsem obecné kroky a rady, kterých bychom se měli při návrhu držet. Také jsem popsal časté problémy, které můžou při návrhu nastat a poskytnul řešení. Jako součást této práce jsem navrhnul jednoduchý fuzzy regulátor pro řízení obráceného kyvadla a popsal kroky, podle kterých jsem ho navrhnul.

2 Teorie řízení

I když možná ne každý by souhlasil, věda a technologie slouží především k tomu, aby lidem usnadnila jejich životy. Inženýři se snaží popsat a modelovat nějakou část reálného světa, kterou nazveme systém, aby pozměnili její chování podle jejich vůle. Řídící systém je potom takový systém, který ovládá činnost jiných prvků nebo systémů. Využívá k tomu jiných prvků pracujících s nějakým signálem, především elektrickým, jako jsou čidla, mikropočítače nebo taky prvky provádějící nějakou fyzickou činnost. Tímto systémem může být například automatický brzdový systém pro automobily, nebo také reaktor jaderné elektrárny [14]. Inženýři se zabývají návrhem a implementací takových systémů.

Návrh takovýchto systémů není jednoduchá činnost, vyžaduje pečlivost, určitou představivost a ochotu návrh několikrát přezkoumat a prověřit jeho správnost. V některých případech je i nutné celý návrh zahodit a začít úplně od začátku [14].

2.1 Řídící systém

Řídící systém je zařízení, které ovlivňuje chování nějakého systému. Vstupem řídícího systému jsou nějaké veličiny, které chceme regulovat. Regulátor tento vstup příjme, a vyšle dál signál. Za signál v regulátoru považujeme většinou nějakou frekvenci elektřiny, která se zasílá většinou po elektrickém vodiči – jde tedy o přenos informace. Tento signál potom v systému může dále řídit nějakou fyzickou činnost, nebo být využit k dalším výpočtům. Prvek co signál převádí na nějakou fyzickou akci nazýváme akční člen, nebo taky akutátor. Obecně systémy dělíme na ty s otevřenou smyčkou, nebo s uzavřenou smyčkou [14].

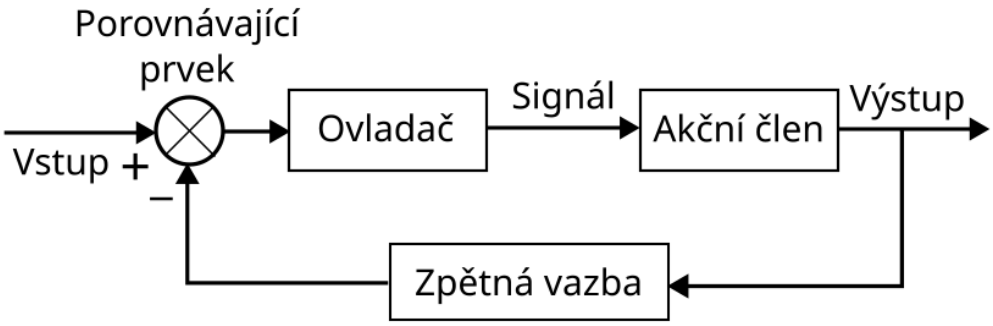
2.1.1 Systémy s otevřenou smyčkou

Tyto systémy nejsou závislé na svém výstupu a jejich výstup řízení nijak neovlivní. Protože takovýto systém nijak nebere v úvahu svůj výstup, nedokáže se vyrovnat s vnějšími vlivy na systém. Příkladem takového systému může být například obyčejný toustovač, u kterého nastavíme jen teplotu ohřevu nebo čas. Za nějakou dobu se toustovač sám vypne, bez ohledu na stav toustu. U takového systému se spoléháme na to, že nebude nijak výrazně ovlivňován vnějšími podmínkami [14].

Obrázek 1: Systém s otevřenou smyčkou

2.1.2 Systémy s uzavřenou smyčkou

Tyto systémy využívají tzv. zpětnou vazbu – pracují se svým výstupem pro zlepšení dalšího řízení. Tyto systémy porovnají svůj aktuální výstup s požadovaným výstupem, a regulaci upraví tak, aby se požadovanému výstupu více blížila. To jim umožní vyrovnat se s vnějšími vlivy působící na systém – například zvýšená teplota, nevhodné povětrnostní podmínky, opotřebení řídících prvků v systému, například senzorů nebo i nějakých akčních členů systému [14]. Vnější vlivy jsou v praxi často nevyhnutelné a musíme s nimi počítat, proto je tento systém praktičtější. Systémy se zpětnou vazbou jsou zastoupeny většinou u velkých průmyslových systémů. Příkladem využití může být cementová rotační pec v Dánsku, sestrojena přibližně v roce 1978 [12]. Tato pec byla řízena tzv. fuzzy systémem, ke kterým se ještě vrátíme a podrobněji je popíšeme.



Obrázek 2: Systém s uzavřenou smyčkou

Vidíme, že z výstupu na konci se signál posílá zpátky ke vstupu, kde se s ním provadí další výpočet.

2.2 Lineární a nelineární systémy

Systémy můžeme taktéž dělit na lineární, nebo nelineární. Lineární systém je takový, jehož signám můžeme modelovat pomocí lineární funkce [3]. Uvažujme nějaký systém, který má vstupy $x_1(t)$ a $x_2(t)$ a výstupy $y_1(t)$ a $y_2(t)$. U systémů uvažujeme i čas t, protože samozřejmě s průběhem času se signál může měnit. Dále předpokládejme, že pro tento systém platí $x_1(t) \to y_1(t)$ a $x_2(t) \to y_2(t)$, tedy že pro vstup $x_1(t)$ nám systém vrátí nějaký výstup $y_1(t)$ pro nějakou konstantu času t, a podobně pro $x_2(t)$ a $y_2(t)$. Aby byl systém lineární, musí splňovat následující dvě podmínky [14][3]:

2.2.0.1 Additivita – výstup pro násobek vstupu musí být stejný jako násobek výstupu pro stejný vstup. Tuto skutečnost můžeme zapsat takto:

$$a \cdot x_1(t) \to a \cdot y_1(t)$$

$$a \cdot x_2(t) \to a \cdot y_2(t)$$

2.2.0.2 Homogenita – výstup součtu dvou vstupů bude stejný jako součet výstupů pro tyto vstupy jednotlivě:

$$x_1(t) + x_2(t) \to y_1(t) + y_2(t)$$

Tyto dvě podmínky můžeme zapsat jako jednu:

$$a \cdot x_1(t) + b \cdot x_2(t) \to a \cdot y_1(t) + b \cdot y_2(t)$$

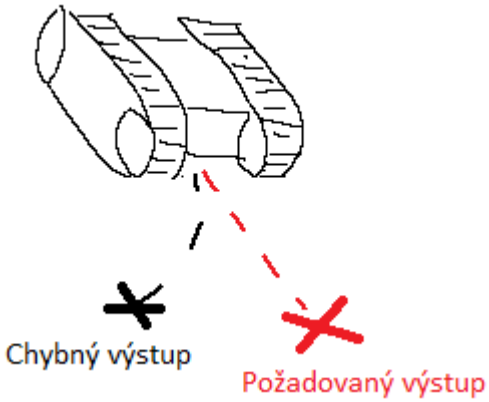
Nelineární systémy jsou všechny systémy, co tyto podmínky nesplňují. Tato oblast je pro nás zajímavější, protože většina systémů v praxi není lineární. Je ale obtížné takové systémy modelovat pomocí běžných postupů. V praxi se většinou

snažíme takový systém přibližně popsat nějakou lineární funkcí, a operovat v nějakém rozumném rozmezí, kde se funkce blíží naší aproximaci. Tento postup užíváme převážně u tzv. PID regulátorů (ovladačů), které jsou v průmyslu dnes velmi rozšířené. Jeden z dalších možných postupů je využít tzv. fuzzy regulátor, který dokáže snadno modelovat nelineární funkce. Oba typy regulátorů si následně popíšeme.

2.3 PID regulátor

Zkratka PID znamená proporcionální, integrační a derivační, což je jméno složek, ze kterých se regulátor skládá. PID regulátor využívá zpětné vazby – porovnává svou aktuální výstupní hodnotu s požadovanou hodnotou a snaží se co nejpřesněji přiblížit [8].

PID regulátor se využívá v systému s uzavřenou smyčkou – například pomocí senzoru systém zjistí, v jakém stavu se nachází. Pomocí této informace zjistí chybu a PID regulátor upraví své chování tak, aby se systém choval uspokojivě. Uvažujme například pojízdného robota řízeného PID regulátorem, který má za úkol dojet na nějakou pozici. Je vybaven dvouma pásy.



Obrázek 3: Robot

Pokud se robotovi do jednoho pásu dostane nějaká nečistota, může se stát, že začne zatáčet na jednu stranu. Pokud je vybaven senzory, porovná svou pozici s požadovanou dráhou a bude se snažit přizpůsobit, pokud je PID regulátor správně nastavený.

Aby PID regulátor fungoval správně, musíme pro něj dobře nastavit konstanty, obvykle pro každou část regulátoru platí jedna konstanta. Podrobněji si strukturu PID regulátoru popíšeme takto [8][16]:

Proporcionální část slouží jako rychlá odezva na změřenou chybu. Regulátor nejdříve odečte změřený aktuální signál od požadovaného a tento rozdíl, kterému říkáme také odchylka, vynásobí konstantou. Pokud je konstanta příliš vysoká, bude docházet k velkým oscilacím – signál se bude nějakou dobu zvyšovat a snižovat, než se ustálí. Pokud naopak bude konstanta příliš nízká, ovladači může trvat déle, než dosáhne požadované hodnoty – bude málo citlivý. V horším případě se signál nemusí vůbec ustálit na námi požadované hodnotě a hodnota bude příliš malá. Když budeme uvažovat $y_P(t)$ jako výstup proporcionálního regulátoru $y(t)$ jako aktuální výstup, $y_s(t)$ jako požadovaný výstup a k jako konstantu, můžeme výstup zapsat takto:

$$y_P(t) = k \cdot (y_s(t) - y(t))$$

Derivační část slouží ke stabilizaci signálu. Snaží se vypočítat velikost chyby v budoucnu, podle toho, jak rychle signál roste v přítomnosti. Derivační regulátor se správně zvolenou konstantou se dokáže rychleji ustálit na požadované hodnotě, což nám může umožnit zvolení agresivnějších konstant a rychlejší celkové odezvy, pokud to systém se kterým pracujeme potřebuje.

$$y_D(t) = k\tau_D \frac{d(y_s(t) - y(t))}{dt}$$

2.3.1 PID regulátory v praxi

PID regulátory jsou jednoduché na nastavení a často používané. Některé zdroje uvádí, že v praxi až 90% regulátorů je právě typu PID, v některých případech ale bez integrační nebo derivační části [8]. V určitých případech se pro lepší řízení systému používá PID regulátor společně s fuzzy regulátorem.

3 Fuzzy regulátor

Fuzzy regulátory pracují na principu fuzzy logiky, které se budeme podrobněji věnovat za malou chvilku. Fuzzy regulátory jsou obecně nelineární. Teoreticky tedy pomocí fuzzy regulátoru dokážeme modelovat jakoukoliv nelineární funkci s libovolnou přesností – jsou tedy univerzálním aproximátorem [19]. Tohle tvrzení bylo dokázáno pomocí Stone–Weierstrassovy věty [17]. Z části právě i díky tomuto chování jsou fuzzy regulátory pro nás zajímavé. Zejména když se snažíme

modelovat nějaký nelineární systém – systém, kde i malá změna nějaké veličiny může silně ovlivnit systémové chování. V takových případech je velmi nepraktické používat PID regulátor, který je svým chováním lineární (výstup regulátoru je přímo úměrný jeho vstupu) [8]. I když by PID regulátor dosáhl požadované hodnoty díky zpětné vazbě, mohl by způsobovat potíže.

Fuzzy regulátory našly své uplatnění v mnoha odvětvích. Jsou využívány například pro běžné domácí spotřebiče, ale i pro počítačové vidění, řízení jaderného reaktoru, ve zdravotnictví a jiných odvětvích. Fuzzy regulátory jsou v některých případech používány společně s PID regulátory. Abychom si mohli podrobněji popsat jak pracuje fuzzy regulátor, musíme nejprve pochopit, co je to fuzzy logika.

3.1 Fuzzy logika

Fuzzy logika oproti booleovské nepracuje jen za pomoci dvou pravdivostních hodnot, ale operuje s celým jejich intervalem – jedná se tedy o typ vícehodnotové logiky. Jako interval pravdivostních hodnot si můžeme zvolit $\langle 0; 1\rangle$, kde 0 je hodnota nepravdivá a 1 je hodnota pravdivá. Tento typ logiky je mimo jiné odvozen od teorie fuzzy množin, která je určitým zobecněním klasických množin [7]. Teorie fuzzy množin byla zavedena profesorem Lotfi A. Zadehem v roce 1965.

3.1.1 Fuzzy množiny

Ze začátku si připomeneme, jakým způsobem zapisujeme klasické (nebo také ostré) množiny. Možný způsob zápisu je obyčejným výčtem prvků: $A = \{a, b, c, d\}$, nebo použitím pravidla, například: $A = (x|x$ je písmeno abecedy$)$.

Další způsob je pomocí funkce, která nám říká, které z prvků do množiny patří nebo nepatří. Tuto funkci nazýváme funkci příslušnosti množiny A nad univerzem X a můžeme ji zapsat takto [1]:

$$\mu_A(x) = \begin{cases} 1 & \text{, když } x \in A \\ 0 & \text{, když } x \notin A \end{cases}$$

U obyčejných množin máme jen prvky, které do nich patří (funkce vrátí 1), nebo nepatří (funkce vrátí 0). Abychom získali fuzzy množinu, stačí tuto funkci poupravit tak, že ji rozšíříme na celý interval reálných hodnot $\langle 0; 1\rangle$ [7]. Zároveň ale musíme definovat i nějaké univerzum, tedy nějakou množinu, nad kterou bude funkce definována. Funkci příslušnosti μ nad univerzem X je funkce $\mu : X \to \langle 0; 1\rangle$. Tuto funkci můžeme ztotožnit se samotnou fuzzy množinou. Fuzzy množinu A tedy můžeme definovat jako funkci [2]:

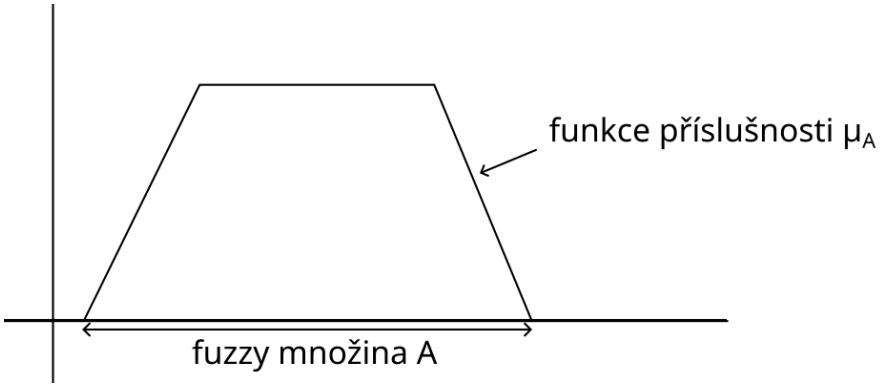
$$A : X \to \langle 0; 1\rangle$$

Fuzzy množinu A můžeme také zapsat jako množinu uspořádaných dvojic prvků univerza X a hodnoty příslušnosti $\mu_A(x)$ pro daný prvek [7]. Dvojice znázorňuje

přiřazení hodnoty příslušnosti $\mu_A(x)$ pro prvek $x \in X$ [6]:

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$

Obě definice přiřazují hodnoty příslušnosti k dané fuzzy množině prvkům nějakého univerza, říkají nám tedy prakticky to samé.



Obrázek 4: Fuzzy množina A s funkcí příslušnosti μ_A

3.2 Operace na fuzzy množinách

Na fuzzy množinách můžeme podobně jako na klasických (kde funkce příslušnosti nabývá hodnot 0 nebo 1) používat množinové operace. Pro definici pravdivostních hodnot těchto operací používáme T-normy a S-normy, které si podrobněji rozepíšeme později. Uvažujme fuzzy množiny A a B. Operace nad fuzzy množinami můžeme definovat takto [4]:

Sjednocení:

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = max(\mu_A(x), \mu_B(x))$$

kde $max(x, y)$ je funkce $max : \langle 0, 1\rangle \times \langle 0, 1\rangle \to \langle 0, 1\rangle$, která vrátí x, pokud $x > y$, jinak vrátí y.

Průnik:

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = min(\mu_A(x), \mu_B(x))$$

kde $min(x, y)$ je funkce $min : \langle 0, 1\rangle \times \langle 0, 1\rangle \to \langle 0, 1\rangle$, která vrátí x, pokud $x < y$, jinak vrátí y.

Vidíme tedy, že sjednocení a průnik dvou fuzzy množin můžeme definovat jako fuzzy disjunkci a fuzzy konjunkci pro nějaký prvek x.

Doplněk:

$$\mu_{\overline{A}}(x) = C(\mu_A(x))$$

kde C definujeme jako funkci $C : \langle 0, 1\rangle \to \langle 0, 1\rangle$, která musí splňovat dvě pravidla:

(1) Platí, že $C(0) = 1$ a $C(1) = 0$.

(2) Pokud máme $a, b \to \langle 0, 1\rangle$ a $a < b$, tak platí, že $C(a) \geq C(b)$.

Operaci C můžeme nazvat fuzzy negací. Příklad takové negace může být třeba $C(x) = 1 - x$. Pokud platí i následující podmínka:

(3) Platí, že $C(C(x)) = x$.

tak je C tzv. silná negace.

3.2.1 T-normy a S-normy

T-norma je funkce dvou proměnných $T : \langle 0, 1\rangle \times \langle 0, 1\rangle \to \langle 0, 1\rangle$, která musí splňovat pro všechna x, y, x', y', $z \in \langle 0; 1\rangle$ tyto následující pravidla [7][4]:

Komutativita: $T(x, y) = T(y, x)$

Asociativita: $T(T(x, y), z) = T(x, T(y, z))$

Monotonnost: Pokud $x \leq x'$ a $y \leq y'$, tak platí, že $T(x, y) \leq T(x', y')$

Okrajové hodnoty: $T(x, 1) = x$ a $T(x, 0) = 0$

Pokud je nějaká funkce T-norma, znamená to, že ji můžeme použít jako fuzzy konjunkci. Po zamyšlení zjistíme, že námi dříve definovaná operace průniku jako minimum dvou hodnot příslušnosti splňuje všechna definovaná pravidla a jedná se tedy také o T-normu [4]. Existuje více T-norm, které bychom mohli použít místo funkce $min(x, y)$, a pár dalších si ještě ukážeme později. Pro obecnou konstrukci fuzzy regulátorů ale není třeba znát všechny.

Podobně pro operaci fuzzy disjunkce používáme operace nazývané S-normy. S-norma je taktéž funkce dvou proměnných $T : \langle 0, 1\rangle \times \langle 0, 1\rangle \to \langle 0, 1\rangle$. Pro všechna x, y, x', y', $z \in \langle 0; 1\rangle$ S-normy musí splňovat následující pravidla [4]:

Komutativita: $S(x, y) = S(y, x)$

Asociativita: $S(S(x, y), z) = S(x, S(y, z))$

Monotonnost: Pokud $x \leq x'$ a $y \leq y'$, tak platí, že $S(x, y) \leq S(x', y')$

Okrajové hodnoty: $S(x, 1) = 1$ a $S(x, 0) = x$

Každá S-norma má vždy svou komplementární T-normu. Pro T-normu min to bude S-norma max. Pro nějaké hodnoty $a, b \in \langle 0, 1\rangle$ můžeme komplementární S-normu vypočítat z T-normy takto [18]:

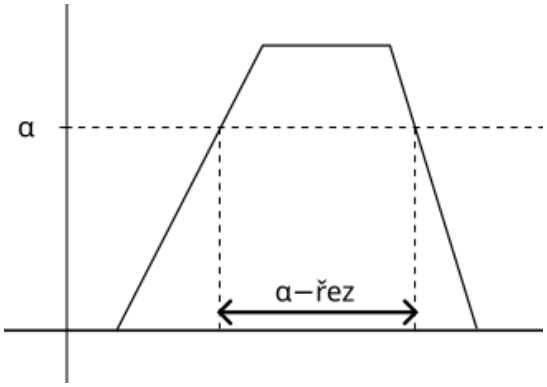
$$S(a, b) = 1 - T(1 - a, 1 - b)$$

3.2.2 α-řez fuzzy množiny

Mějme nějakou fuzzy množinu A nad univerzem U a nějaké α, pro které platí $0 \leq \alpha \leq 1$, potom α-řez K fuzzy množiny A je množina [4]:

$$K_\alpha = \{x \in U \mid \mu_K(x) \geq \alpha\}$$

α-řez fuzzy množiny je tedy ostrá množina obsahující všechny prvky větší nebo rovno α.



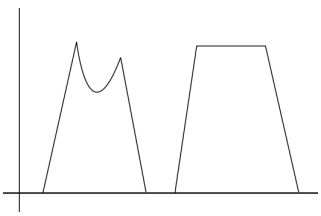
Obrázek 5: α-řez fuzzy množiny

3.2.3 Konvexní fuzzy množina

Fuzzy množina A nad univerzem U je konvexní, pokud platí pro všechna x, $y \in U$ a $a \in \langle 0, 1\rangle$ vzhledem k T-normě T [4]:

$$\mu_A(a \cdot x + (1 - a) \cdot y) \geq min(\mu_A(x), \mu_A(y))$$

Jinak řečeno, funkce příslušnosti konvexní fuzzy množiny by měla být monotónně neklesající do nějaké určité hodnoty a následně monotónně nestoupající. Další možná definice je, že pro libovolný α-řez fuzzy množiny musíme získat vždy jen jeden spojitý interval [7].



Obrázek 6: Nekonvexní fuzzy množina (vlevo), konvexní fuzzy množina (vpravo)

Tabulka 1: Fuzzy relace R: x je finanční fond s riskovým faktorem y [7]

$R =$

	l	m	h
s	0.0	0.3	1.0
i	0.6	0.9	0.1
r	0.8	0.5	0.2

3.3 Fuzzy relace

Relace mohou být použity k definici různých vztahů mezi proměnnými. Než začneme s popisem relací fuzzy množin, připomeneme si relace u obyčejných ostrých množin. Binární relace nad universem U_1 a universem U_2 je podmnožina S kartézského součinu $U_1 \times U_2$. Páry $(u_1, u_2) \in U_1 \times U_2$ z relace S většinou znázorňují nějakou vlastnost [7].

Běžné relace můžeme generalizovat na fuzzy relace. Fuzzy relace jsou užitečné, když chceme analyzovat nějaký vztah mezi vstupními a výstupními hodnotami fuzzy regulátoru. Nejprve definujme kartézský součin dvou fuzzy množin A_1 a A_2 jako [7]:

$$\mu_{A_1 \times A_2}(x, y) = \mu_{A_1}(x) \cap \mu_{A_2}(y)$$

Binární fuzzy relace definujeme podobně jako klasické binární relace. Binární fuzzy relace R je podmnožina kartézského součinu dvou fuzzy množin $A_1 \times A_2$ s funkcí příslušnosti $\varrho_R : U_1 \times U_2 \to \langle 0, 1 \rangle$ nad universem U_1 a universem U_2. Podle velikosti hodnoty příslušnosti $\varrho_R(u_1, u_2)$ určíme sílu relace mezi u_1 a u_2 [7].

Příklad si můžeme ukázat na fuzzy relaci R mezi finančními fondy $X = \{s, i, r\}$ (shares, fixed-interest stocks a real estates) a riskovým faktorem $Y = \{l, m, h\}$ (low, medium a high). Relace ukáže, jak moc velký riskový faktor y má fond x, například pro fond s má riskový faktor h hodnotu 1.0. To můžeme zapsat takto: $\varrho_R(s, h) = 1.0$ [7].

3.3.1 Skládání fuzzy relací

Skládání fuzzy relací funguje obdobně jako skládání obyčejných relací. Mějme dvě fuzzy relace $R \in X \times Y$ a $S \in Y \times Z$. Složením fuzzy relací získáme fuzzy relaci definovanou nad $X \times Z$ s funkcí příslušnosti [7]:

$$(\varrho_S \circ \varrho_R)(x, z) = \bigcup_{y \in Y} \{\varrho_R(x, y) \cap \varrho_S(y, z)\}$$

kde pro $\cup$ a $\cap$ můžeme použít livobolné binární operace, nejčastěji ale používáme kombinaci operátorů maximum pro $\cup$ a minimum pro $\cap$ (tzv. max-min kompozice) nebo maximum pro $\cup$ a násobení pro $\cap$ (tzv. max-product nebo max-dot kompozice). Pro další příklad si zde přesněji definujeme max-min kompozici [7]:

$$(\varrho_S \circ \varrho_R)(x, z) = sup\{min\{\varrho_R(x, y), \varrho_S(y, z)\} \mid y \in Y\}$$

Tabulka 2: Fuzzy relace R': x je finanční fond s možností zisku/ztráty z [7]

$R' =$

	hl	ll	lp	hp
l	0.0	0.4	1.0	0.0
m	0.3	1.0	1.0	0.4
h	1.0	1.0	1.0	1.0

Tabulka 3: Fuzzy relace R': y je riskový faktor s možností zisku/ztráty z [7]

$R' \circ R =$

	hl	ll	lp	hp
s	1.0	1.0	1.0	1.0
i	0.3	0.9	0.9	0.4
r	0.3	0.5	0.8	0.4

Nyní si vzpomeňme ještě jednou na výše uvedenou tabulku znázorňující riskový faktor finančních fondů. Nyní uvažujme další relaci R' nad riskovým faktorem Y a možnost zisku $Z = \{hl, ll, lp, hp\}$ (high loss, low loss, low profit a high profit), která bude určovat možnost zisku vzhledem k riskovému faktoru [7].

Nyní chceme zjistit, jaká bude možnost ztráty nebo zisku pro jednotlivé finanční fondy. Tuto informaci můžeme zjistit složením relací R' a R. Po složení těchto dvou relací pomocí min-max kompozice získáme relaci [7]:

$(\varrho_{R'} \circ \varrho_R)(s, hl) = max\{min(0,0), min(0.3,0.3), min(1,1)\} = 1$
$(\varrho_{R'} \circ \varrho_R)(s, ll) = max\{min(0,0.4), min(0.3,1), min(1,1)\} = 1$
$(\varrho_{R'} \circ \varrho_R)(i, hl) = max\{min(0,0.6), min(0.3,0.9), min(1,0.1)\} = 0.3$
$\vdots$

3.3.2 Relace podobnosti

Relace podobnosti nám pomohou lépe pochopit, jakým způsobem fuzzy regulátory fungují. Dokážeme díky nim také popsat, jak moc jsou si různé hodnoty podobné. Relaci podobnosti E definujeme jako funkci $E : U \times U \to \langle 0, 1\rangle$ s T-normou T nad univerzem U, která pro všechna $x, y, z \in U$ splňuje tyto podmínky [7]:

Reflexivita: $E(x, x) = 1$

Symetrie: $E(x, y) = E(y, x)$

Tranzitivita: $T(E(x, y), E(y, z)) \leq E(x, z)$

Příklad relace podobnosti může být $E(x, y) = 1 - min(|x - y|, 1)$.

3.4 Rozšiřující obal

Prvky ve fuzzy množině, které mají podobné vlastnosti, by měly mít i podobnou příslušnost k fuzzy množině [7]. Této vlastnosti říkáme rozšiřitelnost. Mějme

relaci podobnosti $E : U \times U \to \langle 0, 1\rangle$ s T-normou T nad univerzem U. Fuzzy množinu μ nazveme rošiřitelnou, pokud:

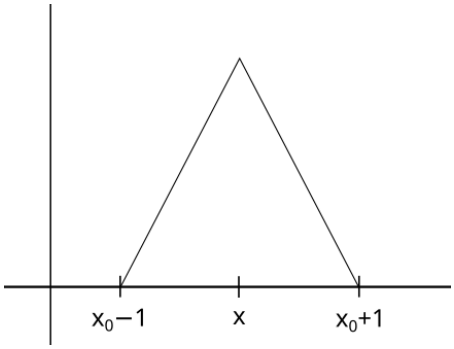
$$T(\mu(x), E(x, y)) \leq \mu(y)$$

platí pro všechna x, $y \in U$.

Ne každá fuzzy množina je rozšiřitelná, ale můžeme ji vždy na rozšiřitelnou předělat. Pomocí následující definice můžeme z fuzzy množiny vytvořit rozšiřitelnou fuzzy množinu [7]:
Mějme relaci podobnosti $E : U \times U \to \langle 0, 1\rangle$ s t-normou T nad univerzem U. Rozšiřující obal $\widehat{\mu}$ fuzzy množiny μ s relací podobnosti E definujeme [7]:

$$\widehat{\mu}(y) = sup\{t(E(x, y), \mu(y))|x \in U\}$$

Ostrou množinu můžeme považovat jako speciální příklad fuzzy množiny, kde všechny prvky mají maximální příslušnost, rozšiřující obal můžeme tedy sestrojit i pro ostré množiny. Pro příklad uvažujme relaci podobnosti $E(x, y) = 1 - min(|x - y|, 1)$ s T-normou $T(x, y) = max(0, x + y - 1)$ (Łukasiewiczova T-norma) [7]. Na obrázku je rozšiřující obal pro jednoprvkovou množinu $\{x_0\}$ vzhledem k relaci E a T-normě T. Tento rozšiřující obal vytvoří trojúhelníko-



Obrázek 7: Rozšiřující obal pro jednoprvkovou množinu x

vou funkci příslušnosti $\widehat{\mu}$. Pomocí rozšiřujícího obalu můžeme definovat funkce příslušnosti jako rozšiřující obaly bodů nebo intervalů.

3.5 Škálový faktor

Podobnost dvou měřených hodnot závisí na jednotce měření. *Dvě vzdálenosti měřené například v kilometrech můžou mít velmi podobné hodnoty nebo být téměř nerozeznatelné, ale pokud tyto hodnoty změříme v milimetrech, budou velmi rozdílné. Podobnost dvou hodnot by neměla záviset na jednotkách, ve kterých ji měříme - výška dvou lidí by měla mít mezi sebou stejný stupeň podobnosti (pro nějakou relaci podobnosti E), nehledě na to jestli je měříme v metrech, centimetrech nebo v palcích* [7]. Pro zachování poměru podobnosti musíme hodnoty naškálovat

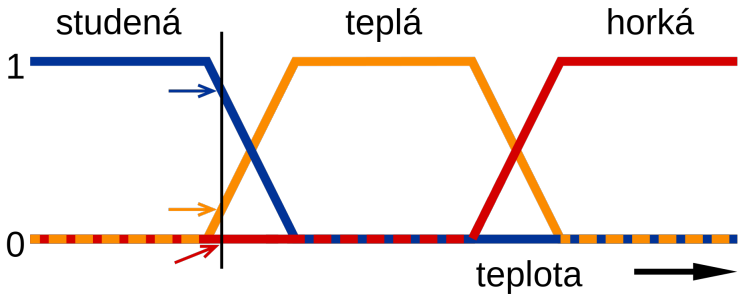
nějakým faktorem $c > 0$. Můžeme si definovat metriku $\rho(x, y) = |c \cdot x - c \cdot y|$, která definuje vzdálenost dvou prvků a využívá škálového faktoru c. Nyní si můžeme definovat relaci podobnosti $E(x, y) = 1 - min(|c \cdot x - c \cdot y|, 1)$ [7].

3.6 Normalizace

Univerzum hodnot pro vstupní a výstupní proměnné fuzzy regulátoru se v různých případech použití mohou lišit. Abychom nemuseli dělat pro každou aplikaci regulátoru manuální změny, můžeme univerzum hodnot normalizovat. Univerzum hodnot normalizovaného ovladače je většinou tvořeno intervalem $\langle -1, 1 \rangle$ (někdy se také používá interval $\langle 0, 1 \rangle$) [19]. Teď už ale ovšem naše vstupní a výstupní hodnoty nemusí patřit do tohoto intervalu. Proto musíme v regulátoru ještě vstupy také normalizovat a výstupy následně denormalizovat. Pro normalizaci vstupů a denormalizaci výstupů musíme vybrat vhodné škálové faktory.

3.7 Lingvistické proměnné

Hodnoty lingvistických (někdy také jazykových) proměnných jsou tvořeny slovy v nějakém jazyce [19]. Pro příklad si teplotu vody T definujeme jako lingvistickou proměnnou. Tato lingvistická proměnná bude moct nabývat hodnot *cold*, *warm*, *hot*. Těmto hodnotám také říkáme termy. Termy jazykových proměnných mají také svůj číselný rozsah - univerzum. Univerzum je množina nějakých číselných hodnot, kterých mohou nabývat - u teploty to může být například interval hodnot Celsiovy stupnice $\langle -5, 120 \rangle$. Nakonec přiřadíme každé z lingvistických hodnot nějakou funkci příslušnosti nad daným univerzem. Termy lingvistických proměnných můžeme znázornit jako fuzzy množiny. Jazykovou proměnnou tedy



Obrázek 8: Termy lingvistických proměnných (studená, teplá a horká) [5]

dohromady tvoří 4 prvky: Její název, množina lingvistických hodnot, univerzum a pravidlo přiřazující funkci příslušnosti k jednotlivým lingvistickým hodnotám [19].

3.7.1 Funkce fuzzy regulátoru

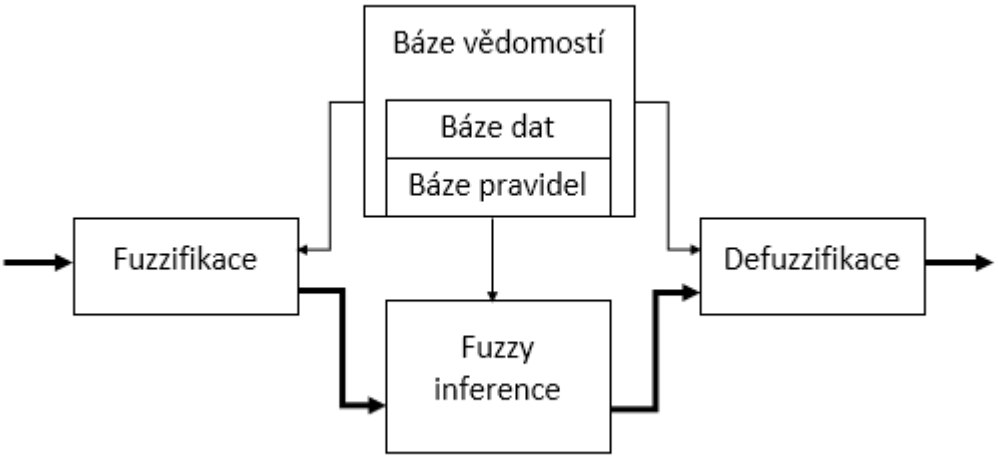
Řízení pomocí fuzzy logiky se v jistém smyslu snaží napodobit způsob, jakým člověk přemýšlí nad provedením nějaké akce. Když člověk řídí auto, neuvažuje v přesných hodnotách, například že rychlost je $27,6$ kilometrů za hodinu. Spíše si řekne, že jede trochu pomalu. Pro člověka není problém odhadnout fyzikální veličny a popsat je pomocí přibližných hodnot [7].

Fuzzy regulátor pracuje v několika krocích. Nejprve fuzzifikátor přiřadí každé vstupní hodnotě stupeň příslušnosti ke vstupním množinám.

Dalším krokem je fuzzy inference. V tomto kroku se na vstupní fuzzy množinu aplikují pravidla. Tímto krokem poté získáme ze vstupních fuzzy množin výstupní fuzzy množiny. Pravidla jsou zapsána ve formě IF-THEN, tedy například [9]:

$$\text{IF } (\textit{speed} \text{ is } \textit{LOW } \textit{AND } \textit{temperature} \text{ is } \textit{HOT}) \text{ THEN } (y \text{ is } \textit{FAST})$$

IF část pravidla se nazývá antedecent a THEN část se nazývá konsekvent.



Obrázek 9: Obecné schéma fuzzy regulátoru

Všechna pravidla fuzzy regulátoru jsou obsažena v bázi pravidel. Inferenční systém rozhodne, která pravidla se aktivují a jak silně. Síle této aktivace říkáme také stupeň aktivace. Podle typu použitého fuzzy systému, inferenční systém může zkombinovat stupně aktivace jednotlivých pravidel spolu s konsekventem pravidla do výstupní fuzzy množiny, nebo může jednotlivě zaslat každý stupeň aktivace pravidla do defuzzifikátoru [9]. Defuzzifikátor tak získá buďto fuzzy množinu, nebo jednotlivé stupně aktivace spolu s informací o fuzzy množině konsekventu pravidla. Obecně tento výstup potom převede do ostrého výstupu, který už bude sloužit k řízení nějakých dalších prvků systému.

3.7.1.1 Využití fuzzy regulátoru

V praxi jsou kontrolní systémy často ovlivněny různými vlivy. Senzory mohou být například ovlivněny šumem, deštěm a jinými povětrnostními podmínkami a vstupy získané výpočetním modelem nemusí být vždy přesné. Dále se také můžeme setkat s jevy působící na systém, které nejsou lehce měřitelné, například jaký má vzduch vliv na letadlo [11]. Ve fuzzy regulátoru jsou tyto odchylky řešeny vhodným výběrem pravidel a fuzzifikátoru.

3.7.1.2 Mamdani a Takagi-Sugeno-Kang fuzzy systémy

Mamdaniho systém byl vyvinutý Ebrahimem H. Mamdanim v roce 1975. V tomto systému jsou antecedenty i konsekventy pravidel zapsány v jazykových proměnných. Už jsme si ukázali, že hodnoty těchto proměnných jsou vlastně fuzzy množiny. Takto může vypadat pravidlo v Mamdaniho fuzzy regulátoru.

$$\text{IF } (speed \text{ is } LOW \ AND \ temperature \text{ is } HOT) \text{ THEN } (y \text{ is } FAST)$$

Takagi-Sugeno-Kang (někdy také jen Takagi-Sugeno) systém byl vyvinut roku 1985. Na rozdíl od Mamdaniho systému jsou konsekventy pravidel tvořeny funkcemi – může tedy být rychlejší, protože nemusí docházet k defuzzifikaci. Uvažujme, že $f(x, y)$ je nějaká polynomiální funkce [7]:

$$\text{IF } (x_1 \text{ is } A_1 \; AND \; x_2 \text{ is } A_2) \text{ THEN } (y \text{ is } Y)$$

3.8 Kdy použít fuzzy regulaci?

Pokud problém nedokážeme řešit běžným způsobem (například pomocí PID regulátoru), je vhodné použít fuzzy ovladač. Nejprve bychom ale měli zmínit důvody, proč může být obyčejný PID regulátor lepší volba. PID ovladače jsou lehké na implementaci, vyžadují méně nastavení (stačí jen tři konstanty) a spousta lidí s nimi má zkušenosti, kdežto fuzzy regulátor potřebuje alespoň nějakou znalost fuzzy logiky. Fuzzy regulátor má také více parametrů k vyladění oproti PID regulátoru. Je také těžší popsat přesný chod regulátoru, a díky tomu může být těžší najít a opravit chyby. Na druhou stranu, fuzzy regulátory jsou často v průmyslu využívány s úspěchem. Může to být díky tomu, že řízení probíhá pomocí IF-THEN pravidel, sestavených za použití běžných slov jako například "málo", "hodně"nebo "zvyšující se". Fuzzy regulátor je vhodné použít, pokud máme lidské experty, kteří dokáží námi řešený proces popsat [4].

4 Návrh fuzzy regulátoru

Fuzzy systémy při návrhu využívají právě nepřesných lidských pojmů. Většinou za využití vědomostí nějakého experta, který nám pomůže ve volbě parametrů regulátoru. Pro návrh obyčejných ovladačů nejprve musíme znát matematický popis systému, ale pro návrh fuzzy systému přesný matematický popis nepotřebujeme. Myšlenka fuzzy regulátorů spočívá právě v zachycení zkušeností experta, který dokáže systém řídit i bez vědomostí o jeho vnitřním chování. Fuzzy ovladač tedy můžeme víceméně navrhnout pomocí své intuice [9].

Následně si ukážeme podrobnější konstrukci Mamdaniho fuzzy regulátoru. Konstrukci fuzzy regulátoru můžeme obecně rozdělit na několik kroků [9][4][13]:

(1) Průzkum informací o systému

(2) Volba vstupních a výstupních proměnných

(3) Rozdělení univerza vstupů a výstupů na fuzzy podmnožiny

(4) Přiřazení funkce příslušnosti ke každé lingivstické proměnné

(5) Sestrojení báze pravidel

(6) Normalizace - Volba vhodných škálových faktorů pro vstupní a výstupní proměnné

(7) Volba fuzzifikační metody pro vstupní proměnné

(8) Výběr metody fuzzy inference

(9) Spojení fuzzy výstupů aktivovaných pravidel do jedné fuzzy množiny

(10) Volba metody defuzzifikace pro získání ostrého výstupu

4.1 Průzkum informací o systému

Než začneme s konstrukcí regulátoru, musíme nejdříve zjistit, jakou funkci má regulátor provádět a vybrat vhodné parametry pro jeho funkci. Pro každý tento parametr musíme definovat vhodné univerzum - nemá smysl, aby například termostat v pokoji počítal i se zbytečně vysokými teplotami. Taktéž se musíme snažit do univerza zahrnout všechny hodnoty, které mohou nastat. Pokud bude náš regulátor často počítat s hodnotami mimo náš zvolený rozsah, nemusí to přinášet uspokojivé výsledky [9].

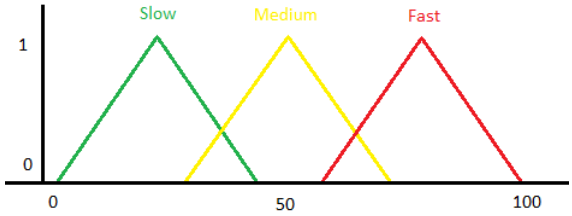
Příkladem tohoto rozsahu může být například již zmiňovaný termostat. Pokud budeme jako vstup považovat aktuální teplotu pokoje, dávalo by smysl jako univerzum uvažovat teploty například mezi 10°C a 40°C. Tyto hodnoty se můžou pro každý systém lišit. V případě, kdy si nevíme rady, je dobré možný rozsah konzultovat s nějakým lidským expertem [9].

S velkým množstvím vstupů se může regulátor stát nepřehledným a bude těžké předpovídat jeho chování, snažíme se tedy počet vstupů omezit. Ve většině případů si vystačíme se dvěmi nebo třemi vstupy, přílišný počet vstupů se může stát překážkou při konstrukci regulátoru [19]. Není to ale pravidlo a pokud pro dobré řízení systému potřebujeme více vstupů, můžeme tak učinit.

4.2 Rozdělení univerza na fuzzy podmnožiny

Nejprve bychom se měli zamyslet, jakých hodnot by měly lingvistické proměnné nabývat. Podle toho poté rozdělíme univerzum této proměnné na fuzzy množiny, obvykle je nejlepší začít u tří fuzzy množin a v případě potřeby jejich počet zvýšit. Ve většině případů se používá počet od tří do devíti fuzzy množin [9]. Například vstupní lingvistické proměnné *speed* můžeme přiřadit hodnoty: $SLOW$, $MEDIUM$, $FAST$.

S větším počtem vstupních fuzzy množin bychom měli definovat více pravidel, proto bychom se měli snažit vybrat rozumný počet fuzzy množin. Při návrhu funkcí příslušnosti pro fuzzy množiny je obecně vhodné, aby jsme pokryli celé universum. Kdyby se na vstupním univerzu vyskytoval nějaký prvek, který do žádné z množin nenáleží, systém by s tímto prvkem nedokázal pracovat, což by bylo nežádoucí [7].



Obrázek 10: Rozdělení univerza na tři fuzzy podmnožiny

Při volbě výstupních fuzzy množin bychom taky měli myslet na to, jakou použijeme defuzzifikační metodu, což podrobněji probereme později.

4.2.1 Výběr funkce příslušnosti

Funkce příslušnosti může nabývat různých tvarů, avšak pro konstrukci fuzzy regulátoru bychom se měli snažit volit jen konvexní fuzzy množiny. Po funkci příslušnosti také chceme, aby alespoň v jednom bodě x nabývala hodnoty $\mu(x) = 1$.

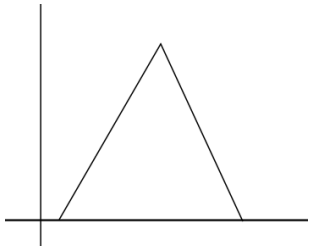
Obecně by tvar funkce měl znázorňovat hodnoty příslušnosti prvků univerza k dané lingvistické proměnné. Většina výzkumů ukazuje, že obecně nejvhodnější tvary funkcí příslušnosti jsou trojúhelníkové a lichoběžníkové funkce. Složitější tvary většinou nevedou k o moc lepším výsledkům a navíc komplikují výpočet. Můžeme ale narazit na situace, kde se nám i komplikovanější tvary funkcí budou hodit.

Dalším faktorem je také šířka funkcí příslušnosti. Užší funkce příslušnosti nám dovolí od sebe vstupy více odlišit, což vede k větší citlivosti samotného regulátoru [9]. Obecně je také dobré, aby se fuzzy množiny překrývaly. Většinou se doporučuje překrytí fuzzy množiny z 25% [13].

4.2.1.1 Trojúhelníková funkce příslušnosti

Jeden z nejobecnějších a nejpoužívanějších tvarů funkce příslušnosti, který by měl ve většině případů postačit [13]. Pokud pomocí tohoto tvaru nezískáváme uspokojivé výsledky, můžeme ho zkusit nahradit nějakým složitějším tvarem (například Gaussovou funkcí). Tvar můžeme také chápat jako rozšiřující obal nějakého bodu [7].
Tento tvar definujeme takto [4]:

$$f(x) = \begin{cases} 0 & \text{, když } x \leq a \\ \frac{x-a}{b-a} & \text{, když } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{, když } b \leq x \leq c \\ 0 & \text{, když } c \leq x \end{cases}$$

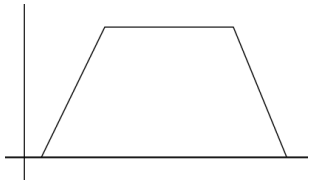


Obrázek 11: Trojúhelníková funkce příslušnosti

4.2.1.2 Lichoběžníková funkce příslušnosti

Tento tvar oproti trojúhelníkovému tvaru funkce popisuje spíše interval hodnot. Tvar můžeme chápat jako rozšiřující obal nějakého intervalu [7]. Tvar definujeme takto [4]:

$$f(x) = \begin{cases} 0 & , \text{když } x \leq a \\ \frac{x-a}{b-a} & , \text{když } a \leq x \leq b \\ 1 & , \text{když } b \leq x \leq c \\ \frac{d-x}{d-c} & , \text{když } c \leq x \leq d \\ 0 & , \text{když } c \leq x \end{cases}$$

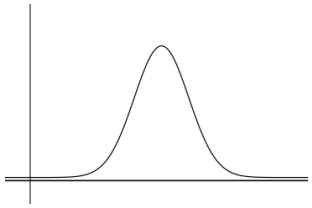


Obrázek 12: Lichoběžníková funkce příslušnosti

4.2.1.3 Gaussova funkce příslušnosti

Tento tvar nabízí jemnější, a možná více přirozené rozprostření stupňů příslušnosti. Výpočetní složitost při použití tohoto tvaru je ale vyšší, měli bychom tedy zvážit jeho použití. Tento tvar definujeme takto [19]:

$$f(x) = e^{\frac{-(x-b)^2}{2c^2}}$$



Obrázek 13: Gaussova funkce příslušnosti

4.3 Sestrojení báze pravidel

Báze pravidel je konečná množina $\mathscr{R}$ IF-THEN pravidel. Uvažujme, že máme i IF-THEN pravidel v množině $\mathscr{R}$. Dále mějme vstupní hodnoty x_1, x_2, ..,x_n, vstupní fuzzy množiny A_i^1, A_i^2, .., A_i^n, výstupní hodnotu y, výstupní fuzzy množinu Y_k, kde k je nějaké očíslování všech výstupních fuzzy množin, spojku AND nebo OR a unární operátor NOT. Pravidlo R_i ve Mamdaniho fuzzy regulátoru za použití nějaké spojky AND zapíšeme touto formou (operátor NOT můžeme libovolně kdekoliv použít, nebo ho i úplně vynechat) [7]:

$$R_i\text{: IF } (x_1 \text{ is } (NOT)\ A_i^1\ AND\ ...\ AND\ x_n \text{ is } (NOT)\ A_i^n) \text{ THEN } (y \text{ is } Y)$$

Obecně pro pravidla můžeme také spojku AND libovolně nahradit spojkou OR. Fuzzy množina pro pravidlo R může být také chápána jako fuzzy relace nad množinami $X_1 \times \cdots \times X_n$ a Y [7].

Výsledek báze pravidel závisí na inferenční metodě, kterou si popíšeme za chvilku. Předpokládáme, že každá část nám přiřadí funkční hodnotu v podobě množiny pro každou kombinaci vstupů $x_1, \ldots, x_n$: Pokud vstupní hodnoty splní předpoklad pro danou část, získáme jednoprvkovou množinu s odpovídající funkční hodnotou. Pokud předpoklad není splňen, získáme prázdnou množinu. Když poté sjednotíme výstupy všech částí, získáme funkční hodnotu celého regulátoru pro vstupy $x_1, \ldots, x_n$ [7]. Celý proces bude dávat větší smysl po pochopení principu agregace, který za chvíli popíšeme.

4.3.1 Jak vytvořit pravidla

Pravidla sestrojujeme podle své intuice za pomocí názorů experta, nebo můžeme využít matematický model procesu, jestli nějaký existuje. Za zmínku stojí také tzv. adaptivní fuzzy regulátor, který pomocí nasbíraných dat a několika parametrů pravidla sám nastaví, zde ho ale probírat nebudeme.

Pravidla by měla být navrhována jedno po jednom. Po navrhnutí všech pravidel bychom měli provést analýzu celé báze pravidel. Je obecně dobré, když báze pravidel splňuje tyto podmínky [9]:

4.3.1.1 Kompletnost

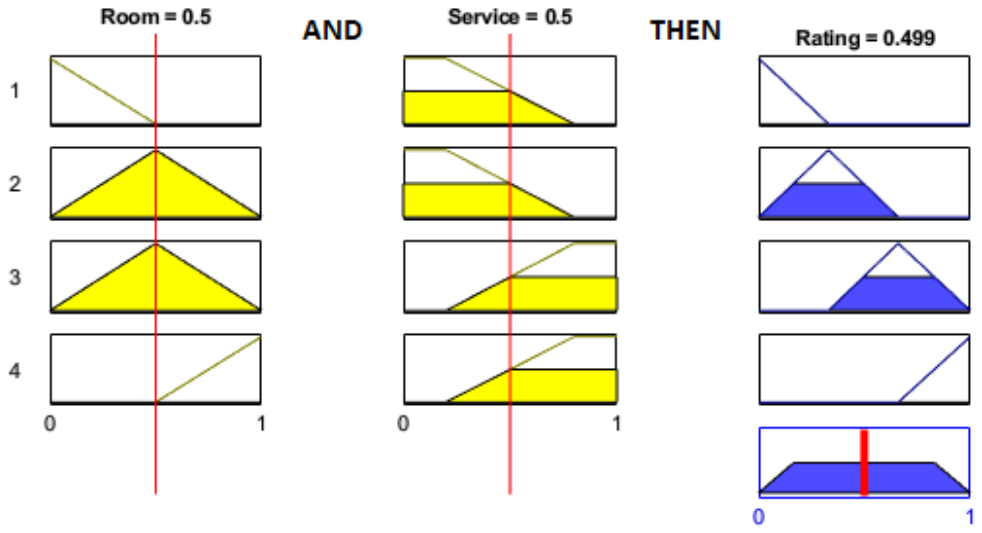
Báze pravidel je kompletní, pokud pro každou kombinaci vstupů $x_1, \ldots, x_n$ existuje nějaký výstup y. *Nemusíme ale formulovat pravidlo pro každou možnou kombinaci fuzzy množin [7].* Mohli bychom také říct, že by se pro každou kombinaci vstupů mělo aktivovat alespoň jedno pravidlo [9].

Pokud by báze pravidel tuto podmínku nesplňovala, mohlo by se stát, že by systém pro určitou kombinaci vstupů neměl žádný výstup, což může vyvolat nežádoucí chování. Jsou ale i případy, kdy nějaká kombinace vstupů v systému nastat nemůže. V těchto případech není nutné pravidlo specifikovat, protože by se v praxi pravidlo stejně nikdy neaktivovalo. Pokud vybereme fuzzy množiny, které se dostatečně překrývají, budeme mít záruku, že se nějaké pravidlo aktivuje i pokud jsme neformulovali pravidlo pro každou kombinaci fuzzy množin.

Uveďme si příklad nekompletní báze pravidel. Budeme se snažit vypočítat hodnocení hotelu podle kvality pokoje a obsluhy. Uvažujme dvě vstupní fuzzy množiny $room = \{bad, standard, great\}$ a $service = \{bad, good\}$ a výstupní fuzzy množinu $rating = \{poor, okay, good, excellent\}$. Pravidla si definujme takto:

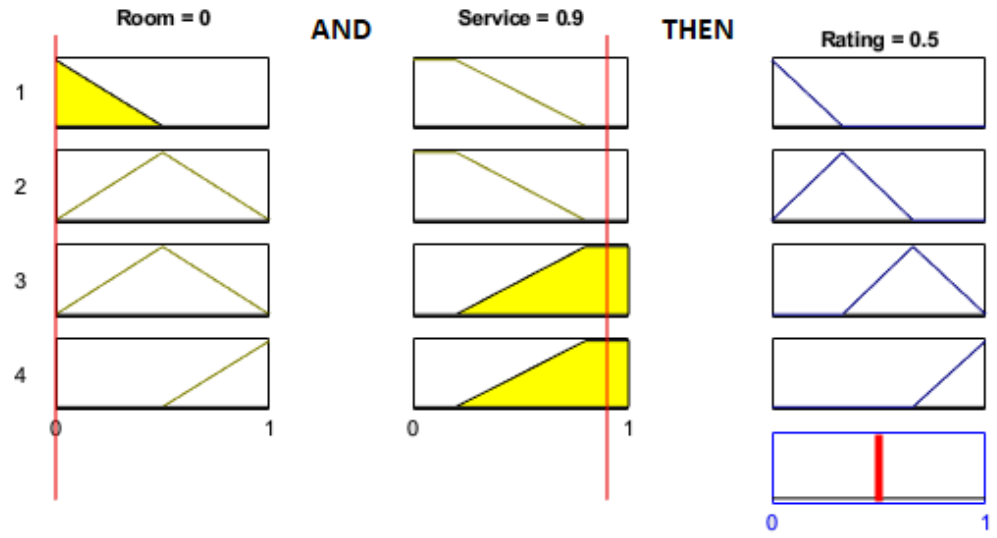
R_1: IF ($room$ is bad AND $service$ is bad) THEN ($rating$ is $poor$)
R_2: IF ($room$ is $standard$ AND $service$ is bad) THEN ($rating$ is $okay$)
R_3: IF ($room$ is $standard$ AND $service$ is $good$) THEN ($rating$ is $good$)
R_4: IF ($room$ is $great$ AND $service$ is $good$) THEN ($rating$ is $excellent$)

Vidíme, že jsme zde nepokryli všechny možné kombinace vstupních fuzzy množin, to samo o sobě ale nemusí být nutně chyba. Problém lépe znázorníme na obrázku: Horizontální čáry značí vstupy, vyplnění trojuhelníku značí příslušnost k dané



Obrázek 14: Znázornění inference

fuzzy množině. Víme, že pravidlo se aktivuje, pokud alespoň trochu vstupy náleží do všech příslušných vstupních lingvistických proměnných. Zde se nám aktivovaly dvě pravidla, nevidíme zde tedy zatím žádný problém, ale uvažujme extrémní případ, kdy se kvalita pokoje bude rovnat nule a kvalita služby blízko jedničce: Jak



Obrázek 15: Znázornění inference nekompletní báze pravidel

můžeme vidět, ani jedno z pravidel se zde neaktivovalo. Tento problém můžeme vyřešit tím, že rozšíříme funkce příslušnosti tak, aby se nějaké pravidlo vždy aktivovalo. Můžeme taktéž například použít Gaussovu funkci příslušnosti, která je nenulová na celém intervalu vstupního univerza. Dalším možným řešením je, pokud nám to systém dovolí, definovat nějakou základní hodnotu pro právě tyto případy. Dalším řešením je přidat více pravidel.

Některé zdroje uvádí, že báze pravidel je kompletní, jen pokud specifikujeme pravidlo pro každou možnou kombinaci vstupních fuzzy množin. Pokud bychom tímto způsobem bázi pravidel navrhnuli, zaručí nám to kompletní bázi pravidel. Pokud ale problém vyžaduje velký počet vstupů, můžeme zbytečně zvyšovat výpočetní složitost a ještě ve výsledku získat nepřehlednou bázi pravidel.

4.3.1.2 Konzistence

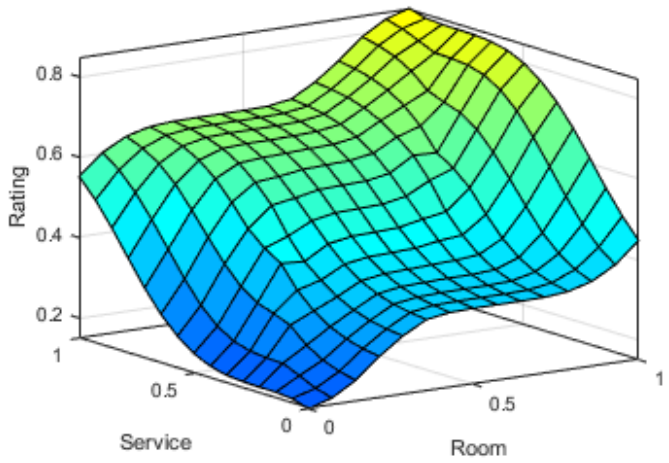
Pro dvě stejné IF části (antedecent) pravidla nesmí být THEN část (konsekvent) pravidla rozdílná. Jednalo by se o případ, kdy pro stejný předpoklad vyvodíme rozdílné závěry. Mějme antedecenty pravidel I_i, I_j a konsekventy pravidel K_i, K_j. Pak musí platit [2][19]:

$$(I_i = I_j) \Rightarrow (K_i = K_j)$$

Obecně tedy pro žádná dvě pravidla nesmí platit, že:

R_1: IF (x_1 is A_1 AND ... AND x_n is A_n) THEN (y is Y_1)
R_2: IF (x_1 is A_1 AND ... AND x_n is A_n) THEN (y is Y_2)

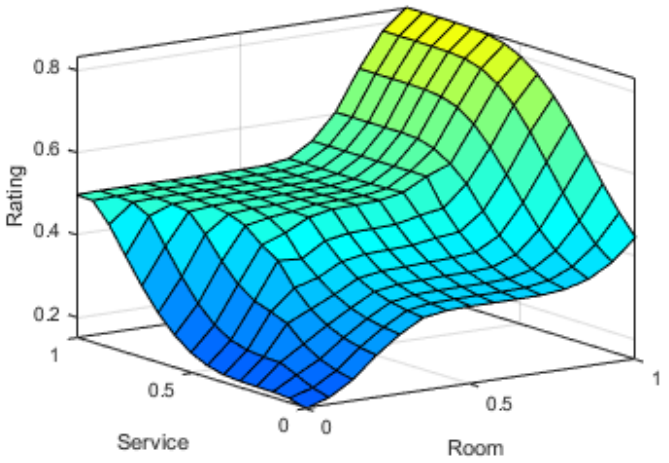
Tyto dvě pravidla by tuto podmínku nesplňovala. Jedno z pravidel bychom tedy měli upravit nebo odstranit. Pro zajímavost si můžeme ukázat, co by se stalo, kdybychom toto pravidlo nedodrželi. Uvažujme opět příklad hodnocení hotelu, který jsme už upravili tak, aby jeho báze pravidel byla kompletní. Chování regulátoru můžeme znázornit touto plochou: Nyní do do báze pravidel přijdeme ještě



Obrázek 16: Znázornění inference konzistentní báze pravidel

jedno pravidlo, které má stejnou IF část jako jedno z pravidel:

R_5: IF ($room$ is $standard$ AND $service$ is $good$) THEN ($rating$ is $okay$)



Obrázek 17: Znázornění inference nekonzistentní báze pravidel

Můžeme vidět, jakým způsobem se plocha chování změnila - je více plochá. Dalo by se říct, že regulátor bude nyní méně rozlišovat mezi různými hodnotami. I když

to ne vždy může být problém, je dobré si na to dát pozor. Pro horší ukázku si dále ještě představme extrémní případ, kdy by se auto chtělo vyhnout překážce.

R_1: IF (*obstacle* is in_front AND *distance* is *small*) THEN (*turn* is $left$)
R_2: IF (*obstacle* is in_front AND *distance* is *small*) THEN (*turn* is *right*)

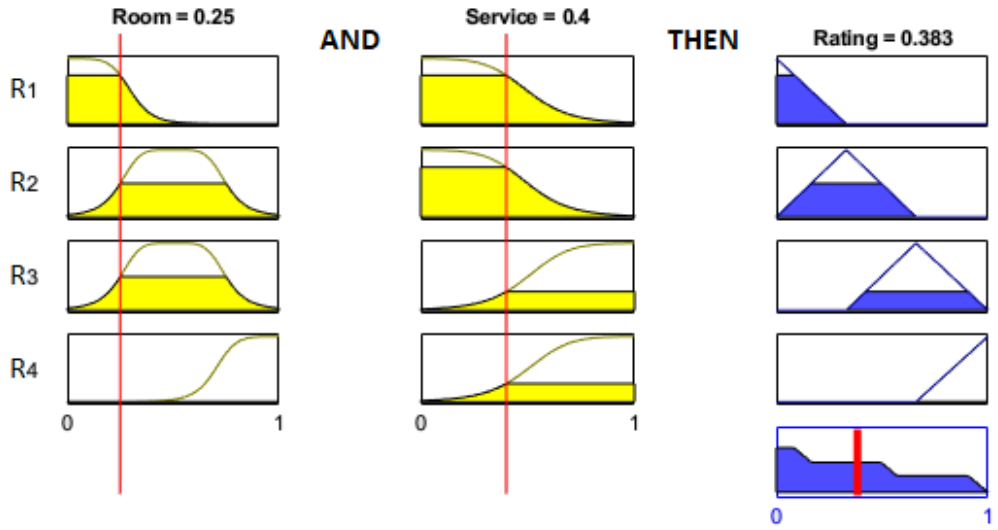
Tímto způsobem bychom auto nasměrovali přímo do překážky, což nejspíš nechceme, pokud neděláme regulátor pro crash-test vozidel.

4.3.1.3 Spojitost

Pokud není prázdný průnik IF částí dvou pravidel a průnik jejich THEN částí není prázdný, tak jsou pravidla spojitá. Báze pravidel je spojitá, pokud jsou libovolná dvě pravidla spojitá. Přesná definice zní takto: Mějme antedecenty pravidel I_1, I_2 a konsekventy pravidel K_1, K_2. Pak pro spojitou bázi pravidel platí [2]:

$$(I_1 \cap I_2 \neq \emptyset) \Rightarrow (K_1 \cap K_2 \neq \emptyset)$$

Uvažujme ještě jednou příklad hodnocení hotelu. Z obrázku lépe uvidíme, proč báze pravidel není spojitá. Vidíme, že pro pravidla R_1 a R_2 spojitost pravidel



Obrázek 18: Znázornění inference nespojité báze pravidel

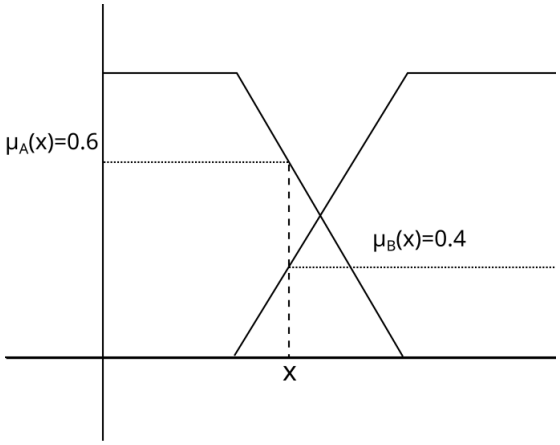
platí, stejně tak i pro pravidla R_2 a R_3, kdežto pravidla R_1 a R_3 mají neprázdný průnik v IF části a prázdný průnik v THEN části. To nám stačí, abychom z podmínky usoudili že tato báze pravidel není spojitá. Spojitost báze pravidel není nutným pravidlem a některé zdroje ji ani jako pravidlo nepovažují, ale zajistí jemné chování regulátoru.

Dalším dobrým pravidlem je, že všechny výstupní fuzzy množiny by měly být obsaženy v THEN části alespoň jednoho pravidla – definovat fuzzy množiny,

které poté nevyužijeme je zbytečná ztráta paměti. Na závěr ještě uveďme, že pokud nebude báze pravidel některé z pravidel splňovat, neznamená to nutně, že regulátor nebude schopný plnit svou funkci. Tyto pravidla nám ale konstrukci kvalitního regulátoru usnadní.

4.4 Fuzzifikace

Fuzzifikace je převod ostrých vstupů na hodnoty příslušnosti k fuzzy množinám dané lingvistické proměnné. Výsledek fuzzifikace prvků z univerza tedy přímo závisí na námi zvolených vstupních fuzzy množinách pro dané univerzum. Fuzzifikace probíhá tak, že pro zvolenou ostrou hodnotu vypočteme hodnoty funcke příslušnosti pro všechny vstupní fuzzy množiny, možná podobně jako výpočet hodnoty obyčejné funkce v nějakém bodě. Z obrázku vidíme, že fuzzifikací hod-



Obrázek 19: Proces fuzzifikace

noty x_0 získáme hodnotu příslušnosti 0.6 k fuzzy množině A a 0.4 k množině B.

4.4.1 Operátor fuzzy implikace

Fuzzy implikaci pro inferenční systém regulátoru můžeme chápat jako funkci $I : \langle 0, 1\rangle \times \langle 0, 1\rangle \to \langle 0, 1\rangle$. Operátor fuzzy implikace nám udává způsob, jakým z antedecentu pravidla vyvodíme konsekvent. V tomto případě tedy nemluvíme o fuzzy implikaci takové, jak bychom si představili v klasické logice. Volba tohoto operátoru je velmi důležitá, protože na něm závisí celý výsledek pravidla. Operátory fuzzy implikace můžeme rozdělit na několik tříd.

4.4.1.1 Funkce implikace

Jedná se o rozšíření implikace z booleovské logiky. Tyto implikace rozlišujeme dále na dvě podtřídy [15]:

S-implikace – někdy také (S,N)-implikace, zavedená podle definice implikace z booleovské logiky $A \Rightarrow B \equiv \neg A \vee B$. Funkci implikace s použitím S-normy S a unární operace silné negace N (například $N(x) = 1-x$) tedy definujme takto:

$$I(x, y) = S(N(x), y)$$

R-implikace – Reziduální (residual) implikace, získaná reziduací zleva spojité T-normy (platí, že: $lim_{x \to {}^{-}a} T(x) = T(a)$). Pro každou T-normu T existuje binární operátor $R : \langle 0, 1\rangle \times \langle 0, 1\rangle \to \langle 0, 1\rangle$ tak, že $T(z, x) \leq y \iff z \leq R(x, y)$. Poté bude platit:

$$R(x, y) = sup\{z \in \langle 0, 1\rangle \mid T(x, z) \leq y\}$$

Tento operátor R nazveme reziduum a můžeme ho použít jako fuzzy implikaci.

Tyto implikace ve fuzzy logice často popírají některé zákony logiky. Uvažujme pravidlo generalizované modus ponens, které je podobné jako klasické modus ponens s rozdílem, že A' se trochu liší od A a že B' se trochu liší od B [6].

$$\frac{\begin{array}{c} x \text{ is } A' \\ \text{IF } x \text{ is } A \text{ THEN } y \text{ is } B \end{array}}{y \text{ is } B'.}$$

Fuzzy množinu B' můžeme vyvodit z A' a fuzzy pravidla zapsaného jako fuzzy relace $R_{A \Rightarrow B}$ takto [6]:

$$\mu_{B'} = \mu_{A'} \circ R_{A \Rightarrow B}$$

Kde operace $\circ$ může značit max-min kompozici fuzzy relací. Nyní mějme fuzzy relaci R pro $High \Rightarrow Low$. Mějme fuzzy množiny:
$High = \{(0, 0), (1000, 0.25), (2000, 0.5), (3000, 0.75), (4000, 1)\}$
$Low = \{(0, 1), (25, 0.75), (50, 0.5), (75, 0.25), (100, 0)\}$
Kde $High$ značí nadmořskou výšku v metrech, a Low procenta kyslíku v atmosféře [6]. Pro operaci implikace použijeme Gödelovu implikaci:

$$\mu_A(x) \Rightarrow \mu_B(y) = \begin{cases} 1 & \text{, když } \mu_A(x) \leq \mu_B(y) \\ \mu_B(y) & \text{, když } \mu_A(x) > \mu_B(y) \end{cases}$$

V tomto konkrétním přkladu bude tedy $A = High$ a $B = Low$. Pak pravdivostní tabulka $High \Rightarrow Low$ bude vypadat takto [6]: Čísla na vertikální ose značí μ_{High} a čísla na horizontální ose značí μ_{Low}. Každý prvek r_{xy} tabulky R je výsledkem implikace $\mu_{High}(x) \Rightarrow \mu_{Low}(y)$. Pokud předpokládáme, že výška je $High$, aplikací modus ponens získáme [6]:

$$\mu = \mu_{High} \circ R$$

Tabulka 4: Pravdivostní tabulka fuzzy implikace $High \Rightarrow Low$ [6]

$R =$

	1	0.75	0.5	0.25	0
0	1	1	1	1	1
0.25	1	1	1	1	0
0.5	1	1	1	0.25	0
0.75	1	1	0.5	0.25	0
1	1	0.75	0.5	0.25	0

$$\mu = \begin{bmatrix} 0 & 0.25 & 0.5 & 0.75 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0.25 & 0 \\ 1 & 1 & 0.5 & 0.25 & 0 \\ 1 & 0.75 & 0.5 & 0.25 & 0 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 1 & 0.75 & 0.5 & 0.25 & 0 \end{bmatrix} = \mu_{Low}$$

Pokud je tedy výška $High$ a platí, že $High \Rightarrow Low$ získáme tím Low, přesně jak bychom očekávali. Zkusme ale uvažovat výšku $Very\ high$:

$$\mu_{Very\ high} = \begin{bmatrix} 0 & 0.06 & 0.25 & 0.56 & 1 \end{bmatrix}$$

tedy druhou mocninu $\mu_{High}(x)$. Nyní aplikujme modus ponens [6]:

$$\mu = \mu_{Very\ High} \circ R$$

$$\mu = \begin{bmatrix} 0 & 0.06 & 0.25 & 0.56 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0.25 & 0 \\ 1 & 1 & 0.5 & 0.25 & 0 \\ 1 & 0.75 & 0.5 & 0.25 & 0 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 1 & 0.75 & 0.5 & 0.25 & 0 \end{bmatrix} = \mu_{Low}$$

Výsledek není identický s druhou mocninou μ_{Low}. Dokázali jsme tedy, že [6]:

$$\frac{\begin{array}{c} x \text{ is } Very\ high \\ \text{IF } x \text{ is } High \text{ THEN } y \text{ is } Low \end{array}}{y \text{ is } Low.}$$

4.4.1.2 T-normy

Jako operátor fuzzy implikace použijeme nějakou T-normu T. Příkladem může být tzv. Mamdaniho implikace [6]:

$$I(x, y) = min(x, y)$$

nebo součinová (product) implikace:

$$I(x, y) = x \cdot y$$

Z pohledu klasické logiky se nejedná o logickou implikaci (nesplňují například $I(0, 1) = 1$), ale ve fuzzy ovladačích se často používají.

4.4.2 Operátor agregace

Pokud chceme do výsledného výpočtu zahrnout výsledky všech pravidel, musíme je nějakým způsobem sjednotit. Agregační funkce je nějaká funkce s n počtem proměnných $A : \langle 0, 1\rangle^n \rightarrow \langle 0, 1\rangle$. Ve většině případů používáme pro agregaci funkci maximum. Pokud máme různé hodnoty příslušnosti pro určitou hodnotu ve výstupní fuzzy množině regulátoru, bude počítat jen s nejvyšším stupňem příslušnosti. To ale znamená, že pokud se například aktivuje pět pravidel se stupněm aktivace 0.6, výsledek bude stejný, jako kdyby se aktivovalo jenom jedno pravidlo se stupněm aktivace 0.6. V takových případech může dávat větší smysl použít pro agregaci například operátor bounded sum [7]:

$$s(x_1, x_2, ..., x_n) = min(1, x_1 + x_2 + ... + x_n)$$

4.5 Inferenční metoda

Ve většine případů se používá inferenční metoda max-min nebo max-product. Metoda max-min pro implikaci použije operaci minimum a pro agregaci funkci maximum (supremum). Uvažujme bázi pravidel $\mathscr{R} = \{R_1, ..., R_r\}$ s pravidly ve formě:

$$R_i\text{: IF } (x_1 \text{ is } A_i^1 \ AND \ ... \ AND \ x_n \text{ is } A_i^n) \text{ THEN } (y \text{ is } Y_k)$$

Při použití spojky AND a T-normy minima získáme fuzzy množinu $\mu_{R_A}(x) = min(\mu_R^1(x_1), ...\mu_R^n(x_n))$ pro antedecent pravidla, a fuzzy množinu $\mu_{R_K}(y)$ pro konsekvent pravidla. Konečná agregace a implikace max-min metody všech výsledků pravidel bude vypadat takto [19]:

$$\mu_{\mathscr{R}} = \sup_{R \in \mathscr{R}} \{min(\ min(\mu_R^1(x_1), ...\mu_R^n(x_n)),\ \mu_R(y))\}$$

Nicméně nevýhoda tohoto typu inference je idempotence maxima a minima. I když se to tak na první pohled nemusí zdát, tato vlastnost může způsobovat potíže. Uvažujme příklad, kdy si kupec chce koupit jeden z domů $x \in \{A, B\}$ podle toho, jak výhodná je cena a jak dobrá je jeho lokace. Uvažujme tyto dva výroky [7]:

$\varphi_1(x)$: Dům x má dobrou lokaci.

$\varphi_2(x)$: Dům x má dobrou cenu.

Pro dům A bude pravdivost výroků $\varphi_1(A) = 0.8$ a $\varphi_2(A) = 0.6$. Pro dům B bude pravdivost výroků $\varphi_1(B) = 0.6$ a $\varphi_2(B) = 0.6$. Kupec se tedy rozhodne na základě toho, jestli $\varphi_1(A) \wedge \varphi_2(A)$ bude větší nebo menší než $\varphi_1(B) \wedge \varphi_2(B)$. Při použití metody minima budou ale oba dva domy stejně dobré, protože $min(0.8, 0.6) = min(0.6, 0.6)$. Pokud místo minima použijeme nějakou neidempotentní T-normu, například Lukasiewiczovu T-normu, vyjde nám $max(0, 0.8 + 0.6 - 1) = max(0, 0.6 + 0.6 - 1)$ což jednoznačně upřednostní dům A [7]. Jsou tedy i určité případy, kdy T-norma minima nestačí.

Metoda max-product taktéž sjednotí všechny výsledky pravidel do jedné množiny pomocí funkce maxima, ale implikace pravidel probíhá pomocí T-normy součinu $T(x, y) = x \cdot y$. Výstupní množina bude vypadat takto [4]:

$$\mu_{\mathscr{R}} = \sup_{R \in \mathscr{R}} \{min(\mu_R^1(x_1), ...\mu_R^n(x_n)) \cdot \mu_R(y)\}$$

Další metodou, která se avšak v praxi moc nepoužívá je metoda inf-implication (nebo také min-implication). Výstupní množina za použití implikačního operátoru I vypadá takto [10]:

$$\mu_{\mathscr{R}} = \inf_{R \in \mathscr{R}} \{I(min(\mu_R^1(x_1), ..., \mu_R^n(x_n)),\ \mu_R(y))\}$$

4.5.1 Normalizace a denormalizace

V zásadě se regulátor bez tohoto kroku obejde, proto ho není potřeba zde zdlouhavě popisovat. V případě, že chceme univerzum normalizovat do intervalu $\langle 0, 1\rangle$ (nebo případně $\langle -1, 1\rangle$), zvolíme vhodný škálový faktor. K tomuto nám může pomoct nějaký optimalizační algoritmus, jinak si budeme muset vystačit se svou intuicí a metodou pokus-omyl. Pokud ale vstupní a výstupní univerzum dobře známe, volba škálových faktorů není složitá záležitost.

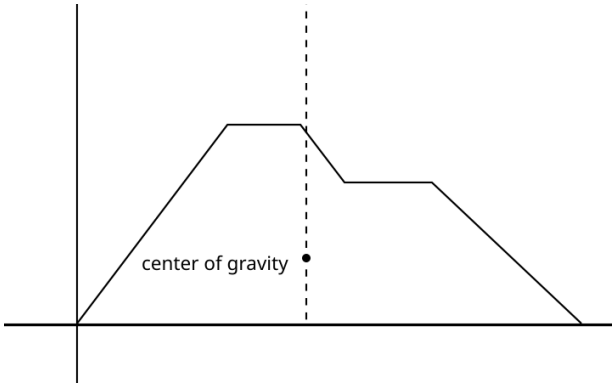
4.6 Defuzzifikace

Po aktivaci pravidel a následné agregaci výstupních fuzzy množin musíme výsledek převést na ostrý výstup. Této akci říkáme defuzzifikace. Existuje spoustu defuzzifikačních metod a několik z nich si zde ukážeme.

4.6.1 Center of gravity

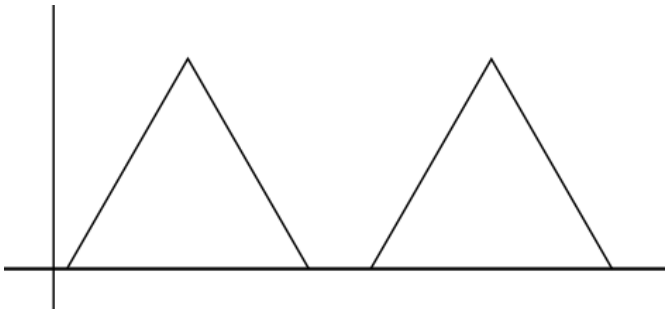
Metoda Center of gravity (nebo také *centroid defuzzification*, zkráceně COG) je často používaná defuzzifikační metoda. Pokud je funkce příslušnosti spojitá, výpočet této metody vypadá takto [7]:

$$y = \frac{\int \mu_{\mathscr{R}}(x) \cdot x \, dx}{\int \mu_{\mathscr{R}}(x) \, dx}$$



Obrázek 20: Grafické znázornění defuzzifikace pomocí metody COG

Nevýhodou této metody je ale poměrně vysoká výpočetní náročnost. Může se také stát, že v některých případech tato metoda nebude vracet výsledek, který bychom očekávali. Uvažujme případ, kdy jako výstup získáme fuzzy množinu složenou ze dvou disjunktních množin [7]. V takovémto případě by metoda COG vypočítala hodnotu někde uprostřed výstupního univerza, kde pravdivostní hodnota fuzzy množiny nabývá nuly. Opět si můžeme představit nežádoucí případ, kdy bychom chtěli pomocí tohoto regulátoru řídit auto. Správná defuzzifikační



Obrázek 21: Výstupní fuzzy množina složená ze dvou disjunktních fuzzy množin

metoda by měla splnit dva úkoly. Měla by proměnit fuzzy množinu na ostrou množinu, a měla by zvolit jen jednu hodnotu z (fuzzy) množiny. Není ale úplně jasné, v jakém pořadí by se tyto úkoly měly provést. Většina defuzzifikačních metod už předem počítá s tím, že získá jen jednu hodnotu a ne množinu hodnot [7].

Dalším problémem je, že za použití metody COG nebudeme schopni získat okrajové hodnoty intervalu výstupního univerza. To znamená, že regulátor nedosáhne minimální nebo maximální výstupní hodnoty. Tento problém můžeme vyřešit rozšířením fuzzy množin za hranice tohoto intervalu, ale musíme poté pohlídat, aby regulátor nevracel hodnoty mimo interval výstupního univerza [7].

4.6.2 Mean of maxima

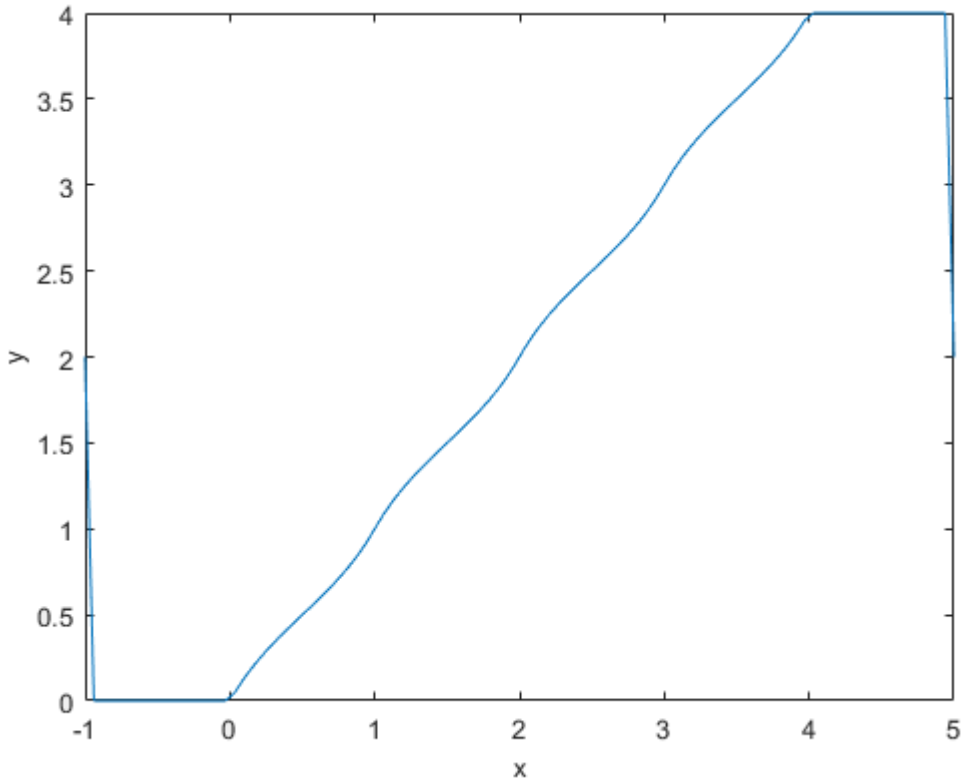
Metoda Mean of maxima (zkráceně MOM) je velmi jednoduchá metoda, která vybere průměr ze všech hodnot s nejvyšším stupňem příslušnosti k výstupní fuzzy množině. Výstupní hodnota takto závisí čistě na pravidlu s nejvyšším stupněm aktivace, pokud nenastala situace, kdy se několik pravidel aktivuje se stejnou sílou. V praxi se tato metoda příliš nepoužívá [7].

Metoda MOM trpí stejným problémem jako metoda COG, pro dvě disjunktní fuzzy množiny by vrátila hodnotu někde uprostřed. Možné řešení problému je zvolit vždy nejvíce pravou (nebo nejvíce levou) nejvyšší hodnotu do výstupní fuzzy množiny [7].

Další nepříjemností metody COG jsou špatné interpolační vlastnoti. Uvažujme případ, kdy se chceme přiblížit funkci $f(x) = x$ pomocí Mamdaniho fuzzy ovladače. Použijeme trojúhelníkové funkce příslušnosti. Zvolme si tedy následující bázi pravidel [7]:

R_1: IF x is *about* 0 THEN y is *about* 0
R_2: IF x is *about* 1 THEN y is *about* 1
R_3: IF x is *about* 2 THEN y is *about* 2
R_4: IF x is *about* 3 THEN y is *about* 3
R_5: IF x is *about* 4 THEN y is *about* 4

Kde fuzzy množina *about* 0 bude mít funkci příslušnosti $\mu_0(x) = 1 - min(|x|, 1)$, fuzzy množina *about* 1 $\mu_1(x) = 1 - min(|x - 1|, 1)$ a podobně. Jak můžeme vidět na obrázku, interpolace není příliš přesná, což není žádoucí.



Obrázek 22: Interpolace funkce $f(x) = x$ pomocí Mamdaniho ovladače

4.7 Takagi-Sugeno-Kang systém

Tento typ systému nepoužívá výstupní fuzzy množiny, pravidla nám vracejí přímo ostré hodnoty. Pravidla v Takagi-Sugeno-Kang ovladači vypadají takto [7]:

$$R\text{: IF } x_1 \text{ is } \mu_R^1 \; AND \; ... \; AND \; x_n \text{ is } \mu_R^n \text{ THEN } y = f_R(x_1, ..., x_n)$$

V konsekventu pravidla můžeme použít i konstantní funkce. Myšlenka toho to regulátoru je najít funkci s pro nás výhodným chováním vzhledem k IF části pravidla. Výsledek výpočtu tohoto typu regulátoru tedy vypadá takto:

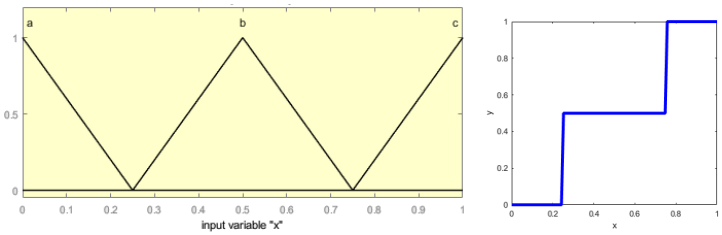
$$y = \frac{\sum_R (\mu_{R,a_1,..,a_n} \cdot f_R(x_1, ..., x_n))}{\sum_R \mu_{R,a_1,..,a_n}}$$

Kde a_1,...,a_n jsou změřené hodonty pro vstupní proměnné x_1,...,x_n, a $\mu_{R,a_1,..,a_n}$ udává sílu aktivace pravidla R.

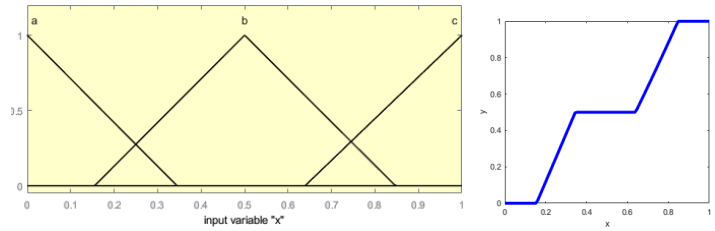
Srozumitelnost Takagi-Sugeno-Kang systému závisí z části na tom, jak moc se vstupní fuzzy množiny protínají. Pokud se vstupní množiny nebudou protínat vůbec, výsledek regulátoru bude vždy funkce, které nám pravidlo přiřadí k danému vstupu. Pokud se funkce budou překrývat, průběh regulátoru bude hladší. Uvažujme bázi pravidel:

$$R_1\text{: IF } x \text{ is } \textit{about } 0 \text{ THEN } y = 0$$
$$R_2\text{: IF } x \text{ is } \textit{about } 1 \text{ THEN } y = 1$$
$$R_3\text{: IF } x \text{ is } \textit{about } 2 \text{ THEN } y = 2$$

Pokud budeme uvažovat vstupní fuzzy množiny na obrázku vlevo, výsledek regulátoru bude takový: Pokud se fuzzy množiny překrývají, výstup bude o něco



hladší: Pokud bychom funkce postupně překrývali více, výstup regulátoru by



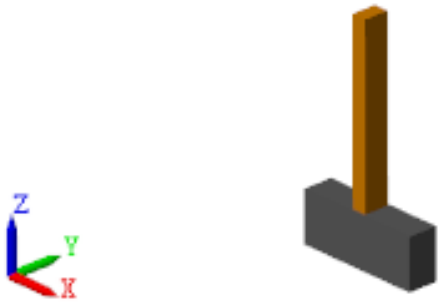
tvořila lineární funkce $y = x$ pro daný interval $\langle 0, 1 \rangle$. Vidíme tedy, že překrytí vstupních fuzzy množin nám umožní poněkud hladší výstup.

4.8 Příklad konstrukce fuzzy regulátoru

Pro ukázku zde ukážu mé řešení konstrukce fuzzy regulátoru pro vyrovnání obráceného kyvadla. Systém kyvadla i regulátor jsem vymodeloval za použití programu MATLAB a jeho nadstavby Simulink.

4.9 Popis systému

Systém se bude skládat z vozíku, který se bude pohybovat jen dopředu nebo dozadu, podél jedné osy. Na horní části tohoto vozíku bude kloubem připojeno kyvadlo. Pomocí pohybu vozíku se bude regulátor snažit zajistit, aby kyvadlo zůstalo vzpřímené a aby byl vozík po vyrovnání ve startovním bodě.



Obrázek 23: Vozík s kyvadlem

4.10 Volba vstupních proměnných

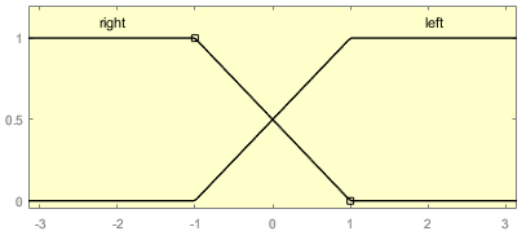
Abychom dokázali kyvadlo vyrovnat, budeme potřebovat nějakým způsobem snímat jeho chování. Abychom kyvadlo udrželi ve vzpřímené poloze, určitě budeme potřebovat znát úhel jeho naklonění. Dalším užitečným parametrem bude úhlová rychlost kyvadla. Představme si případ, kdy má kyvadlo velkou úhlovou rychlost směrem k bodu, kde je vzpřímené. Pokud je kyvadlo již téměř vzpřímené, je vhodné aby kyvadlo zpomalilo, jinak se kyvadlo přetočí na druhou stranu. Tyto dva parametry nám postačí k tomu, aby se kyvadlo vrátilo do vzpřímené polohy a zůstalo v ní.

Další problém, který musíme vyřešit, je aby se vozík s kyvadlem navrátil zpět do původního místa. K tomuto budeme potřebovat dva parametry - rychlost vozíku, a pozici vozíku. Pomocí rychlosti vozíku určíme směr, kterým míří. Pokud například víme, že vozík jede doprava, a nachází se vlevo od startovní pozice, není nutné nic dělat. Pokud ale vozík jede doprava a nachází se vpravo od startovní pozice, musíme zařídit, aby vozík jel doleva.

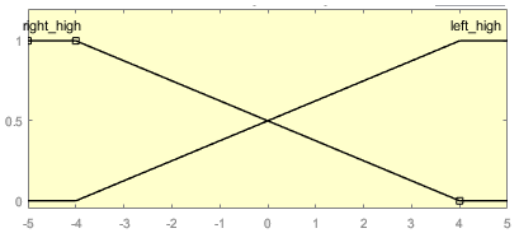
Z předchozí analýzy tedy vychází, že budeme potřebovat čtyři vstupní proměnné: *p_speed* (rychlost kyvadla), *p_angle* (úhel náklonu kyvadla), *c_pos* (pozice vozíku) a *c_speed* (rychlost vozíku).

4.11 Volba funkcí příslušnosti

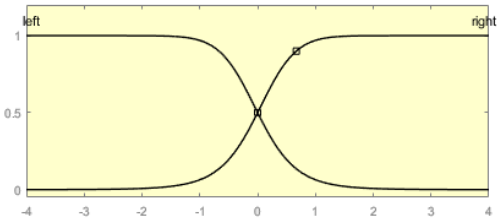
Univerzum každé vstupní lingvistické proměnné nyní pokryjeme fuzzy množinami. Pro fuzzy množiny jsem zvolil velmi jednoduché tvary funkcí příslušnosti. Protože MATLAB ve kterém jsem pracoval sám hodnoty přesahující vstupní univerzum zkrátí, není třeba specifikovat široké univerzum například u pozice nebo rychlosti vozíku.



Obrázek 24: Vstupní proměnná pro úhel náklonu kyvadla



Obrázek 25: Vstupní proměnná pro rychlost kyvadla

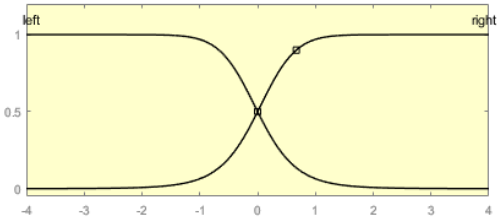


Obrázek 26: Vstupní proměnná pro polohu vozíku

Funkce příslušnosti pro polohu a rychlost vozíku jsem vybral takto, abych zaručil kompletnost vzhledem k mé bázi pravidel.

4.12 Tvorba pravidel

Pravidla vytvoříme pomocí naší intuice. Abychom kyvadlo vyrovnali ve vzpřímené pozici a navrátili vozík na startovní pozici nám postačí 4 pravidla. Mějme tedy čtyři vstupní proměnné: p_speed, p_angle, c_pos, c_speed a jednu výstupní proměnnou c_force. Aby se kyvadlo vzpřímilo, postačí nám tyto dvě pravidla:



Obrázek 27: Vstupní proměnná pro rychlost vozíku

R_1: IF p_speed is $left$ AND p_angle is $left$ THEN c_force is $high\_left$
R_2: IF p_speed is $right$ AND p_angle is $right$ THEN c_force is $high\_right$

Prakticky si to můžeme představit tak, že když se kyvadlo naklání na nějakou stranu, posuneme vozík tím stejným směrem a kyvadlo tím vyrovnáme. Dále abychom vozík přesunuli na startovní pozici, použijeme další dvě pravidla:

R_1: IF c_speed is $left$ AND c_pos is $left$ THEN c_force is low_left
R_2: IF c_speed is $right$ AND c_pos is $right$ THEN c_force is low_right

Tohle pravidlo se může zdát nesmyslné, protože se může zdát, že tak vozík ještě více oddálíme od požadované startovní pozice. Jenže kdybychom vozík jedoucí od startovní pozice doprava postrčili doleva, vozík bychom ještě více zrychlili špatným směrem. Vozík by se totiž snažil vyrovnat kyvadlo, které je nakloněné směrem od středu. Pokud ho ale trošku postrčíme ve stejném směru, kyvadlo se převrátí a snaha kyvadlo vyrovnat ho povede směrem ke startovní pozici.

4.13 Volba inferenční metody

Jako inferenční metodu jsem zvolil základní metodu max-min.

4.14 Volba defuzzifikační metody

Jako defuzzifikační metodu jsem zvolil metodu center of gravity, protože vracela v mém případě uspokojivé výsledky.

Závěr

Cílem mé práce bylo popsat funkci fuzzy regulátoru a poskytnout návod, jak takový regulátor dobře navrhnout. Zezačátku jsem popsal řídící systémy a definoval pojem PID regulátor, ke kterému je fuzzy regulátor alternativou.

Popsal jsem matematické definice fuzzy logiky a fuzzy množin, na kterých tento regulátor stojí. Později jsem předložil kroky a pravidla, kterých bychom se měli při návrhu fuzzy regulátoru držet. Záměrem práce bylo věnovat se především Mamdaniho fuzzy regulátoru, krátce jsem ale i popsal regulátor typu Takagi-Sugeno-Kang. Popsal jsem nežádoucí situace, které mohou nastat a poskytnul řešení. V práci jsem chtěl zmínit i možnost použití fuzzy implikace ve fuzzy regulátorech, nenalezl jsem ale dostatek informací, abych vyvodil jednoznačný závěr.

Na konci práce jsem popsal svůj myšlenkový pochod při sestrojování fuzzy regulátoru pro vozítko s obráceným kyvadlem. K sestrojení jsem použil nadstavby Simulink a Fuzzy logic toolbox programu MATLAB. Ke konstrukci fuzzy regulátoru mi stačila pouze čtyři pravidla a s výsledkem jsem spokojený.

Na závěr bych chtěl říct, že mě těší, kolik jsem se toho dozvěděl o fuzzy regulátorech. Myslím si, že zadání bakalářské práce jsem splnil.

5 Obsah přiloženého datového média

`doc/`
: Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

`data/`
: Soubory pro MATLAB obsahující fuzzy regulátor a systém obráceného kyvadla na vozíku.