

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA ELEKTROTECHNICKÁ



**Diplomová práce**

**Návrh samoučících se regulátorů**

Praha, 2009

Autor: Otto Cerman

## **Prohlášení**

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

V Praze dne \_\_\_\_\_

\_\_\_\_\_ podpis

## **Poděkování**

Děkuji vedoucímu diplomové práce Ing. Petru Huškovi, Ph.D. za jeho čas a spolupráci.

# Abstrakt

V mnoha situacích mohou být složité systémy úspěšně řízeny použitím fuzzy řídicích systémů. Fuzzy regulátor založený na teorii fuzzy logiky a fuzzy množin je úspěšný nástroj pro konverzi lingvistické řídicí strategie ze znalosti experta do báze pravidel regulátoru. Nalézt tato pravidla však vyžaduje mnoho času, protože se často do jisté míry hledají metodou *pokus-omyl*. Navíc pro velmi složité systémy jsou také obtížně zjistitelná.

K tomu převážná většina procesů má proměnnou dynamiku. Z tohoto důvodu v takovýchto situacích klasické regulátory s pevně nastavenými parametry většinou nestačí, protože při změnách procesu je klasické řízení neoptimální a může docházet např. ke ztrátě materiálu nebo energie. V takovýchto případech je potřeba použít adaptivní regulátor.

Samoorganizující se (v angl. psané lit. *self-organizing*) nebo samoučící se (*self-learning*) regulátor patří do skupiny adaptivních fuzzy regulátorů. Jedná se o heuristický regulátor, ve kterém jsou pravidla řízení generována a automaticky vylepšována. Jeho základní funkce jsou: 1. generovat vhodné akční zásahy podle zhodnocení chování systému; 2. modifikovat vhodným způsobem regulátor (nejčastěji bázi znalostí), aby bylo dosaženo požadovaného chování i při změnách pracovních bodů systému.

Cílem této práce je seznámit se s doposud známými metodami, vhodné z nich vybrat a následně je odzkoušet na několika modelech.

# **Abstract**

In many situations complex systems can be controlled successfully using fuzzy control systems. Fuzzy controller based on theory of fuzzy logic and fuzzy sets is successful tool for converting of linguistic control strategy from knowledge of an expert into the rule base of controller. Unfortunately, a lot of time is needed for finding these bases because up to the point they are often searched by trial-error method. In addition they are found out with difficulty for very complex systems too.

Most of plants have variable dynamics. For that reason classical controllers with fixed parameters are not sufficient because during variations of plant classical control systems are not optimal and situations like loss of material or energy can occur. In these cases use of adaptive controller is needed.

Self-organizing or self-learning controllers belong to the group of adaptive fuzzy controllers. They can be seen as heuristic controllers in which control rules are generated and automatically improved. Their basic functions are: 1. to generate suitable control signals according to evaluation of system behaviour 2. to modify controller (most often knowledge base) in a suitable way to reach the required behaviour even during changes of operating points of systems.

The aim of this work is to study known methods, to choose suitable ones and subsequently to test them on several models.

# Obsah

|                  |   |           |
|------------------|---|-----------|
| <b>1</b>         | <b>Úvod</b>   | <b>1</b>  |
| <b>2</b>         | <b>Fuzzy řízení</b>   | <b>2</b>  |
| 2.1              | O principech fuzzy logiky . . . . .   | 2         |
| 2.2              | Fuzzy množiny . . . . .   | 2         |
| 2.3              | Fuzzy řízení . . . . .  | 4         |
| <b>3</b>         | <b>Adaptivní řízení</b>   | <b>8</b>  |
| 3.1              | Co je to <i>adaptivita</i> a proč se používá . . . . .                      | 8         |
| 3.2              | Adaptivní systémy . . . . .   | 8         |
| 3.3              | Adaptivní řízení . . . . .  | 10        |
| <b>4</b>         | <b>Samoučící se regulátory založené na teorii adaptivního fuzzy řízení</b>  | <b>12</b> |
| 4.1              | Samoorganizující se fuzzy regulátor . . . . .                               | 12        |
| 4.1.1            | Použití SOC na konkrétním systému . . . . .                                 | 17        |
| 4.2              | Adaptivní fuzzy regulátor na principu učení s referenčním modelem . . . . . | 19        |
| 4.2.1            | Fuzzy regulátor . . . . .   | 21        |
| 4.2.2            | Referenční model . . . . .  | 23        |
| 4.2.3            | Učící se mechanismus . . . . .  | 24        |
| <b>5</b>         | <b>FMRCLC na konkrétních modelech</b>                                       | <b>30</b> |
| 5.1              | Řízení servosystému . . . . .   | 30        |
| 5.1.1            | Výsledky simulací servosystému . . . . .                                    | 31        |
| 5.2              | Řízení magnetické levitace . . . . .  | 35        |
| 5.2.1            | Výsledky simulací řízení magnetické levitace . . . . .                      | 36        |
| 5.3              | Řízení inverzního kyvadla . . . . .   | 40        |
| 5.3.1            | Výsledky simulací řízení inverzního kyvadla . . . . .                       | 41        |
| <b>6</b>         | <b>Závěr</b>  | <b>49</b> |
| <b>Reference</b> |   | <b>50</b> |

# 1 Úvod

V současné době je známá celá řada způsobů návrhu řídicích systémů, od jednoduchých až po velmi složité návrhy. *Klasické metody* automatického řízení obvykle požadují co možná nejpřesnější znalost mezi vstupy a výstupy, vycházejí tedy výhradně z popisu vnějšího. V 60. letech 20. století vzniká pojem „*moderní teorie řízení*“, která je naopak založena na stavovém popisu. V současné době se oba přístupy (stavový a vnější) silně prolínají. Je-li přesně specifikován model, tak inženýrskou úlohu návrhu regulátoru lze formulovat jako úlohu optimalizační. Místo přímého nastavování konstant regulátoru tedy člověk nastavuje parametry kritéria optimality a případně omezení. Tento přístup je výhodný především v tom, že některé vlastnosti získaných řešení (např. stabilita při kvadraticky optimálním řízení) jsou zajištěny implicitně.

Klasická i moderní teorie řízení tedy vyžadují znalost matematického popisu. U reálných procesů však často bývá velmi obtížné takový popis najít, protože např. není možné provést dostatečný počet měření, měření jsou příliš drahá nebo zdlouhavá. V reálných soustavách navíc čidla, měřící přístroje apod., poskytující data o řízené soustavě, vykazují nepřesnost měření, která je zanášena do procesů dalšího zpracování naměřených údajů, a může tak dojít k absurdní situaci, kdy se snažíme přesně zpracovávat nepřesná data. Výsledný popis řízené soustavy navíc může být natolik složitý, že může být velmi obtížné navrhnut kvalitní regulátor.

Spoléhání na přesný model tedy s sebou přináší i některé nedostatky moderní teorie řízení, které motivují i nové směry výzkumu. Jedná se o tzv. *robustní metody*, které dovolují zahrnout do formulace problému i neurčitost matematického modelu. Pak se ale nutně přistupuje k různým zjednodušením a výsledky už nemusí být uspokojivé.

Robustní metody nám tedy mohou v jistých situacích pomoci, ale existují případy, kdy matematický popis z nějakého důvodu zkrátka nemáme a žádné tyto metody prostě nemůžeme použít. V praxi takové procesy zpravidla reguluje člověk, který ze zkušenosti ví, jak proces řídit, aniž by k tomu potřeboval znát jeho matematický popis. V takovéto situaci se nabízí použití *fuzzy regulátoru*. Nastavování fuzzy regulátorů je však mnohem komplikovanější než nastavování klasických regulátorů. Jeho kvalitní seřízení vyžaduje mnohem více času než nastavení klasického regulátoru, protože dosud neexistuje obecná metoda pro jeho nastavení (pokud pomineme možnost nastavení genetickými algoritmy).

Z předchozích odstavců vyplývá, že by byly velmi vhodné takové metody syntézy regulátoru, které by dokázaly samy dobré nastavit regulátor bez znalosti matematického popisu řízené soustavy v průběhu samotné regulace. Metody řízení, které ho nastaví (popř. postupně stále vyládají) podle změny chování řízeného systému, se nazývají *adaptivní metody řízení*. Klasická teorie adaptivního řízení ale opět vychází ze znalosti matematického popisu.

Jako řešení se tedy nabízí použití nějakého *adaptivního fuzzy regulátoru*, který by byl schopen se sám „naučit“ řídit daný systém pouze na základě hrubých znalostí o řízeném systému, které by měl v počátečním stavu nainicializované. Taková metoda by se tak dala jistě použít i pro vylepšení vlastností již navrhutého „pevného“ fuzzy regulátoru. Uvedené metody však zatím nejsou dostatečně rozpracovány, aby se mohly běžně používat. Cílem této práce<sup>1</sup> je právě tyto metody odzkoušet, popř. vylepšit.

---

<sup>1</sup>Text byl vysázen typografickým systémem L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

## 2 Fuzzy řízení

### 2.1 O principech fuzzy logiky

Slovo „fuzzy“ značí v angličtině něco nejasného, rozmazaného (tj. neostrého), mlhavého. Po řadě pokusů o výstižný překlad se ve většině jazyků ujal termín *fuzzy*, u nějž alespoň nehrozí jiná interpretace. Naopak v anglicky mluvících zemích bývá přílastek „fuzzy“ vnímán jako poněkud negativní, takže jeho použití na výrobcích bývá potlačeno a nahrazováno jinými popisy.

*Fuzzy logika* je matematická disciplína, která si získala značnou popularitu koncem 80. a na začátku 90. let 20. století pro fascinující aplikace uskutečněné především v Japonsku a později i v dalších zemích. Nejúspěšnější aplikace jsou v řízení a regulaci. Hlavním zdrojem úspěchu je to, že fuzzy logika umožnuje zahrnout nepřesnost a poměrně jednoduchým způsobem pracovat s významy slov přirozeného jazyka. Fuzzy logika nabídla překvapivě jednoduchá řešení, která dobře fungují.

Fuzzy regulátory jsou obecně robustnější vzhledem k náhodným poruchám a nepředvídaným situacím než regulátory založené na znalosti matematického popisu řízeného systému. Kromě toho existují úlohy, které lze v současné době řešit pouze pomocí fuzzy regulátoru. Příkladem může být řízení čističky odpadních vod, jejíž dílčí části lze sice řídit klasicky, avšak nad celkem musí dohlížet člověk, který se řídí zkušeností vyjadřitelnou pravidly typu „je-li situace taková, pak udělej následující“. Právě taková pravidla lze pomocí fuzzy logiky implementovat a celý proces tak automatizovat.

Přirozený jazyk hraje mimořádně důležitou roli při libovolné lidské činnosti a umožňuje nám popsat a pracovat s překvapivě konkrétními věcmi. Např. chceme-li se naučit řídit auto, potřebujeme učitele, který nás vede a vysvětluje (v přirozeném jazyce), co máme dělat („nyní jed'te pomaleji, nebrzděte tolík, tady to je nebezpečné“, apod.). Všechny tyto příkazy jsou velmi vágní, a přesto nám stačí k tomu, abychom se řídit naučili.

Neurčitost má (nejméně) dvě stránky: vágnost a nejistotu. Vágnost lze modelovat pomocí *teorie fuzzy množin*, zatímco nejistotu pomocí *teorie pravděpodobnosti* a popř. další teorie, jako je teorie možnosti, různé míry věrohodnosti apod. Zjednodušeně můžeme říci, že pravděpodobnost nám odpovídá na otázku, zda „něco nastane“, zatímco teorie fuzzy množin nám odpovídá na otázku, „co se vlastně stalo“?

Podrobný matematický výklad není předmětem této práce, navíc existuje velká spousta dostupných publikací, které se fuzzy logikou zabývají. Z české literatury je to především [1], [2], v menší míře pak [3]. Každá tato publikace se také věnuje tzv. *fuzzy množinami*, které s fuzzy logikou přímo souvisí.

### 2.2 Fuzzy množiny

Klasická teorie množin poskytla matematice formalismus úspěšně použitý k popisu řady konkrétních situací a rozsáhlému zobecnění, které dovolilo jednotný přístup k řešení řady problémů. Vyskytly se však situace, které tento přístup nedovoluje postihnout obvyklým způsobem. Patří mezi ně i paradoxy, kterými se zabývali už řečtí filosofové, např. paradox hromady: Mějme hromadu kamení a ubírejme kameny jeden po druhém. Otázka je, ve kterém okamžiku přestaneme mít hromadu? Dosud bylo běžné modelovat vágní pojmy pomocí klasických množin, a tedy hraniční prvky musely být zařazeny buď do množiny, nebo mimo ni. Ukazuje se, že v tom je hlavní příčina tak

často kritizované neadekvátnosti matematických metod v praxi, zejména při popisu systémů, jejichž součástí je člověk.

V roce 1965 publikoval L.A. Zadeh článek, kterým zahájil mohutný rozvoj modifikované teorie množin, tzv. *teorie fuzzy množin*. Je to prostředek, který umožní matematicky popsat vágní pojmy a pracovat s nimi. Základním pojmem této teorie je právě pojem *fuzzy množiny*.

Základní myšlenka *fuzzy množin* je jednoduchá a poměrně přirozená. Pokud nejsme schopni stanovit přesné hranice třídy vymezené vágním pojmem, nahradíme rozhodnutí, zda daný prvek do této třídy patří či nikoli, mírou vybíranou z předem definované škály. Každý prvek bude mít přiřazenou míru, která vyjadřuje jeho místo a roli v této třídě. Bude-li škála uspořádaná, pak menší míra bude vyjadřovat, že daný prvek leží na okraji třídy. Tuto míru nazýváme *stupněm příslušnosti*  $\mu$  daného prvku k dané třídě. Třída, v níž každý prvek je charakterizován stupněm příslušnosti k této třídě, se nazývá *fuzzy množina*. Cílem této práce není opět podrobný výklad teorie *fuzzy množin*. Z české literatury je teorie *fuzzy množin* např. dobře zpracována v dostupné publikaci [1].

Zde se spokojíme pouze s intuitivním chápáním pojmu *množina* a uvedeme pouze nezbytně nutné pojmy, které jsou potřeba pro další pochopení. Je známé, že ke každé množině  $A$  existuje množina všech jejích podmnožin. Zato však neexistuje množina všech množin, neboť takový pojem vede ke sporu. Těmto problémům se snadno vyhneeme tak, že se omezíme na studium podmnožin jedné (libovolné, ale pevně dané) tzv. *univerzální množiny* (*univerza*), která se většinou značí symbolem  $X$ .

**Klasické množiny** lze popsat např. výčtem prvků, pravidlem, které musí prvky množiny splňovat, charakteristickou funkcí apod.

Klasická množina:

$$\mu_A : X \rightarrow \{0, 1\} \quad (1)$$

**Fuzzy množiny** jsou zobecněním klasických množin v tom smyslu, že jsou zobrazením na celý interval  $\langle 0, 1 \rangle$ . Jelikož fuzzy množiny obecně nelze popsat jinak než charakteristickou funkcí (tj. funkcí příslušnosti  $\mu$ ), považují se termíny *fuzzy množina* a charakteristická funkce za stejně. Stupeň příslušnosti prvku  $x$  k množině  $A$  označujeme jako  $\mu_A(x)$ .

Fuzzy množina:

$$\mu_A : X \rightarrow \langle 0, 1 \rangle \quad (2)$$

Formálně fuzzy množinu  $A$  zapisujeme častěji jako:

$$A = \{x, \mu_A(x) | x \in X\} \quad (3)$$

Stupeň příslušnosti tedy vyjadřuje „stupeň našeho přesvědčení“, že daný prvek patří do dané fuzzy množiny. Stejně jako pro klasické množiny jsou i pro fuzzy množiny definovány základní operace. Pro účely fuzzy řízení je žádoucí se zmínit o doplňku, průniku, sjednocení a implikaci na fuzzy množinách.

Předpokládejme tedy, že máme dány fuzzy množiny  $A = \{x, \mu_A(x) | x \in X\}$  a  $B = \{x, \mu_B(x) | x \in X\}$  s funkcemi příslušnosti  $\mu_A(x)$  a  $\mu_B(x)$ .

Funkce příslušnosti  $\mu_{\bar{A}}(x)$  *fuzzy komplementu (doplňku)* lze definovat několika způsoby. Jedna z možností, která se používá velmi často, je následující:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (4)$$

Podobně funkce příslušnosti *fuzzy průniku*  $\mu_{A \cap B}(x)$  a *fuzzy sjednocení*  $\mu_{A \cup B}(x)$  lze definovat následovně:

$$\begin{aligned} \mu_{A \cap B}(x) &= \min(\mu_A(x), \mu_B(x)) \\ \mu_{A \cup B}(x) &= \max(\mu_A(x), \mu_B(x)) \end{aligned} \quad (5)$$

Druhý způsob jak definovat průnik  $\mu_{A \cap B}(x)$  je přes algebraický součin, který je definován následovně:

$$\mu_{A \cdot B}(x) = \mu_A(x) \cdot \mu_B(x) \quad (6)$$

Operace fuzzy implikace bude vysvětlena v následující kapitole, ve které bude hrát její výpočet důležitou roli.

## 2.3 Fuzzy řízení

Jedna z možných struktur fuzzy regulátoru je na obr.1. Struktury fuzzy regulátoru se mohou na první pohled trochu lišit. Je to dáno např. tím, že autoři někdy popisují jednotlivé části regulátoru jinými názvy nebo dostatečně jasně nerozlišují mezi bází pravidel a bází znalostí. Nejčastěji používaná konvence ale je, že báze pravidel a funkce příslušnosti (tj. báze dat) tvoří dohromady celek, který se nazývá báze znalostí. Systém (tj. proces) je s regulátorem spojen přes A/D a D/A převodník. V celé této práci budeme pracovat pouze s fuzzy PD regulátorem, proto se následující výklad bude týkat pouze tohoto regulátoru. Vstupem do něho je odchylka  $e(k)$  a její diference  $ce(k)$ , které jsou definovány následovně:

$$e(k) = r(k) - y(k) \quad (7)$$

$$ce(k) = \frac{e(k) - e(k-1)}{T}, \quad (8)$$

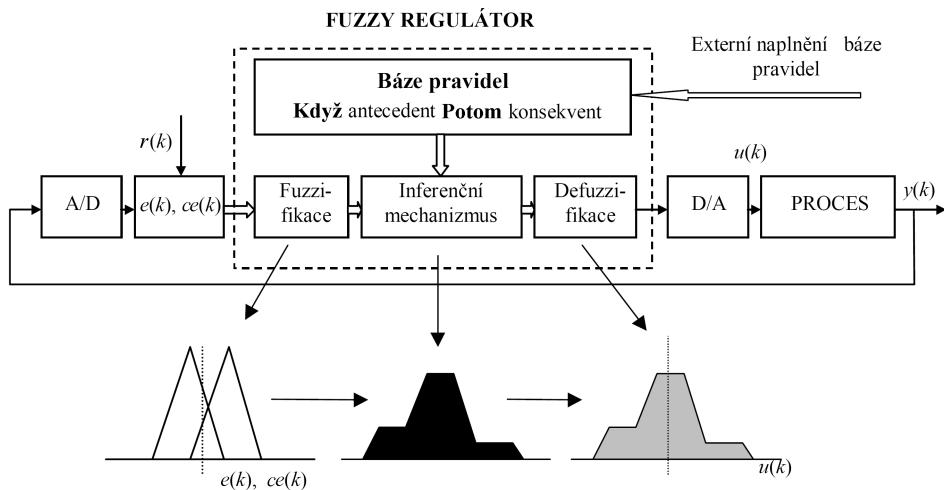
kde  $r(k)$  je žádaná hodnota veličiny,  $y(k)$  je výstupní veličina a  $T$  je perioda vzorkování.

Fuzzy regulátor nejdříve provádí tzv. *normalizaci*. Tato operace spočívá v tom, že různé proměnné, jejichž hodnoty se pohybují v různých rozsazích transformujeme tak, aby všechny nabývaly hodnot z normalizovaného univerza, např. z intervalu  $[-1,1]$ ,  $[0,1]$  a podobně. Další krok se nazývá zpravidla *fuzzifikace*, který spočívá v tom, že každé ostré hodnotě z normalizovaného univerza přiřadíme stupeň příslušnosti do jedné nebo více fuzzy množin odpovídajících významu proměnných použitých v pravidlech. K tomuto účelu musíme pokrýt beze zbytku normalizované univerzum nosiči příslušných fuzzy množin. Blok normalizace však nemusí být ve fuzzy regulátoru vždy použit. V takovém případě je postup stejný, pouze nepracujeme na určeném intervalu hodnot a prvním krokem se pak stává až fuzzifikace.

Pravidla fuzzy PD regulátoru jsou v následující formě:

**Když**  $e(k)$  je  $E^j$  **A**  $ce(k)$  je  $C^l$  **Potom**  $u(k)$  je  $U^m$

$E^j$  a  $C^l$  určuje  $j$ -tou a  $l$ -tou fuzzy množinu asociovanou se vstupem  $e(k)$ , respektivě  $ce(k)$ .  $U^m$  určuje  $m$ -tou fuzzy množinu, která odpovídá výstupu regulátoru  $u(k)$ . Poznamenejme, že se také velmi často výrazy *když, a, jinak, potom* nahrazují anglickými výrazy *if, and, else, then*.



Obrázek 1: Fuzzy PD regulátor ve zpětnovazebním zapojení

Zavedeme značení: N=záporný, P=kladný; S=malý, M=střední, B=velký; ZO=nula a dále jejich kombinaci NB=záporný velký, NM=záporný střední, NS=záporný malý, PS=kladný malý, PM=kladný střední a PB=kladný velký. Několik konkrétních vybraných možných pravidel ve fuzzy PD regulátoru je pak ukázano zde:

**Když**  $e(k)$  je ZO **a**  $ce(k)$  je PS **Potom**  $u(k)$  je PS  
**Když**  $e(k)$  je PS **a**  $ce(k)$  je PS **Potom**  $u(k)$  je PM  
**Když**  $e(k)$  je PS **a**  $ce(k)$  je ZO **Potom**  $u(k)$  je PS  
**Když**  $e(k)$  je ZO **a**  $ce(k)$  je ZO **Potom**  $u(k)$  je ZO

Např. první pravidlo z uvedených předchozích čtyř proto interpretujeme jako:

**Když** odchylka je nula **A** diference odchylky je kladná malá **Potom** akční zásah je kladný malý

Toto je pouze část pravidel, která slouží pro ilustraci. Při 7 funkcí příslušnosti na každém ze dvou vstupů dostáváme celkem  $7 * 7 = 49$  pravidel. V běžné praxi je nejčastěji používán regulátor se dvěma vstupy, protože při více vstupech extrémně narůstá velikost báze pravidel. Otázkou je pak právě to, jak takovou bázi pravidel (popř. bázi znalostí) naplnit. Pokud přidáme ještě jeden vstup, dostaneme se ze 49 pravidel na  $7 * 7 * 7 = 343$  pravidel.

Abychom měli kompletní bázi znalostí, tak musíme kromě báze pravidel ještě zvolit tvar a rozmístění funkcí příslušnosti. Často je pro přehlednost označena identifikátorem a její hodnota je vždy z intervalu  $\langle 0, 1 \rangle$ .

Máme k dispozici celou řadu typů funkcí příslušnosti. Obvykle jsou však používány trojúhelníkové, gaussovské a lichoběžníkové, které nejsou tolik náročné na výpočet jako

obecné spojité funkce a dovolují efektivně využít pamět. Daleko větší vliv než tvar funkcí má však jejich rozmístění. Obecně může být rozložení funkcí lineární (a symetrické) nebo nelineární (a nesymetrické) v jednom či více vstupech a na výstupu. U výstupní proměnné můžeme nelineární rozložení funkcí příslušnosti využívat pro nelineární změny akčního zásahu. V některých aplikacích stupeň nesymetrie může hrát významnou roli.

Vstupní proměnné se tedy v prvním kroku převedou do fuzzy množin. Fuzzifikovaná hodnota pak vstupuje do inferenčního mechanismu, který pracuje s bází pravidel, která společně s funkcemi příslušnosti tvoří bázi znalostí. Těmto krokům se říká *fuzzy inference*.

Inferencí se rozumí celá metoda usuzování. Výsledkem inferenčního mechanismu (prakticky se jedná o vyhodnocení spojek A, Potom, Jinak) je fuzzy množina, ze které je v závěrečném kroku určena ostrá hodnota akčního zásahu  $u$ . Používají se nejčastěji dvě metody fuzzy inference, je to metoda *Min - Max* a metoda *Prod - Max*. Proces výpočtu inference Min - Max části pravidel uvedené na začátku sekce 2.3 je vidět na obr. 2.

Inferenci Min-Max definujeme jako:

$$\mu_{cl}(u) = \max_{l=1}^M (\min(\mu_E(e(k)), \mu_C(ce(k)), \mu_U(u(k)))), \quad (9)$$

kde  $M$  je počet pravidel.

Posledním krokem je převést výstupní data zpět na ostrá data. Tomuto procesu se říká *defuzzifikace*, která také významně ovlivňuje celkový výsledek. K dispozici máme opět celou řadu defuzzifikačních metod. Poznamenejme ještě, že podobně jako k bloku fuzzifikace, tak i k bloku defuzzifikace se zpravidla přiřazuje blok, který provádí denormalizaci akčních veličin.

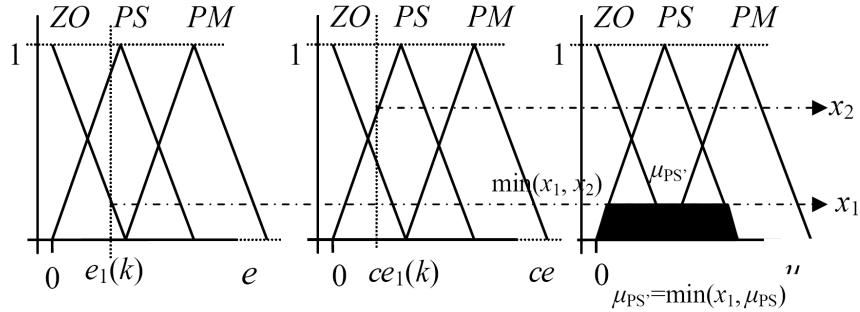
V této práci je použita defuzzifikační metoda těžiště (též metoda centroidů COA - Center of Area). Je to nejznámější a nejčastěji užívaná metoda. Defuzzifikovaná ostrá hodnota odpovídá souřadnici těžiště plochy tvořené fuzzy množinou  $\mu_{cl}(u)$ , která je výsledkem zmíněného inferenčního mechanismu. Těžiště plochy se počítá formálně stejně jako statistická střední hodnota v souboru. Defuzzifikovanou výstupní hodnotu metodou COA v diskrétním případě pak spočítáme podle vztahu:

$$u = \frac{\sum_i u_i \mu_{cl}(u_i)}{\sum_i \mu_{cl}(u_i)} \quad (10)$$

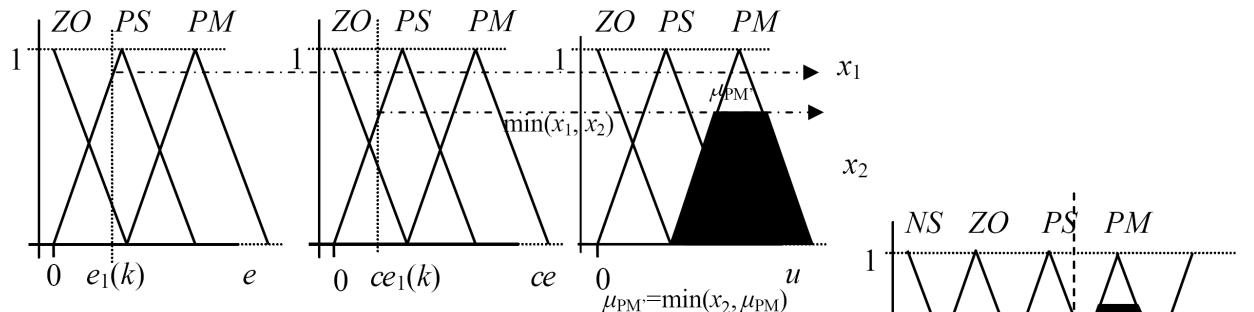
**Předchozí kroky můžeme shrnout do následujících pěti kroků:**

1. fuzzifikaci
2. spojení v pravidlech antecedentu - spojka **A (AND)**
3. spojení antecedentu a konsekventu - implikace **Potom (THEN)**
4. spojení v konsekventu - **Jinak (ELSE)**
5. defuzzifikace

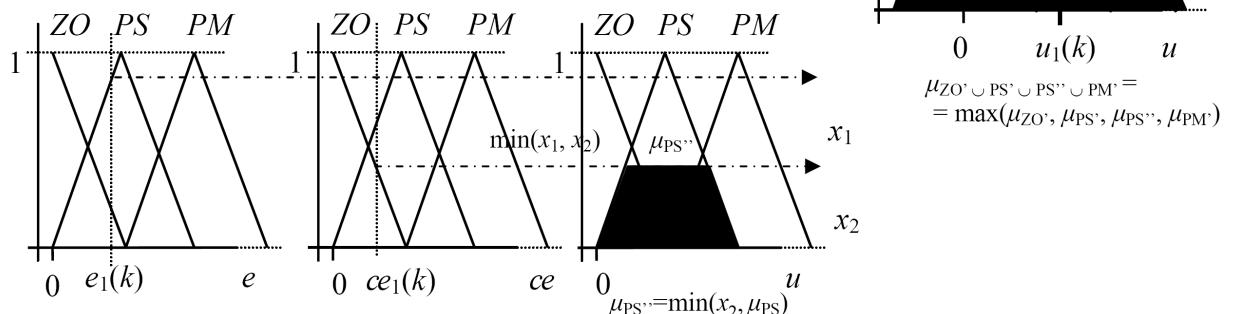
**Když**  $e(k)$  je ZO    A     $ce(k)$  je PS **Potom**  $u(k)$  je PS



**Když**  $e(k)$  je PS    A     $ce(k)$  je PS **Potom**  $u(k)$  je PM

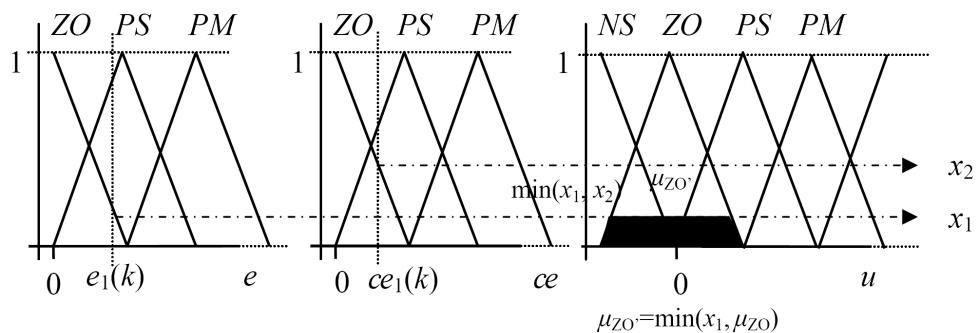


**Když**  $e(k)$  je PS    A     $ce(k)$  je ZO **Potom**  $u(k)$  je PS



$$\mu_{ZO'} \cup PS' \cup PS'' \cup PM' = \\ = \max(\mu_{ZO'}, \mu_{PS'}, \mu_{PS''}, \mu_{PM'})$$

**Když**  $e(k)$  je ZO    A     $ce(k)$  je ZO **Potom**  $u(k)$  je ZO



Obrázek 2: Ukázka výpočtu inference Max - Min, výsledkem je fuzzy množina na obrázku úplně vpravo (obr. poupraven z [4])

## 3 Adaptivní řízení

### 3.1 Co je to *adaptivita* a proč se používá

Převážná většina procesů, se kterými se v průmyslové praxi setkáváme, má proměnnou dynamiku. Změna parametrů procesu je způsobena změnami v provozních režimech (např. různé pracovní výkony), změnami vlastností surovin, paliva, zařízení (opotřebení..). Klasické regulátory s pevně nastavenými parametry často nevyhovují pro řízení takových procesů, protože při změnách procesu je řízení neoptimální a dochází ke ztrátám materiálu, energie, dále k většímu opotřebení, které má pak za následek snižování životnosti zařízení atd. S takovými situacemi se pevně nastavené regulátory často nemohou vyrovnat. Jednou z možností zvýšení kvality řízení takových procesů, je použití adaptivních řídicích systémů.

Adaptace byl původně jev vlastní živým organismům, u kterých se také projevuje v nejpestřejší podobě. Je to vlastnost přizpůsobovat svoje chování změnám okolního prostředí, i když jsou nepříznivé. Každá adaptace představuje pro organismus jistou ztrátu, především energii. Živé organismy při mnohonásobném opakování adaptace na určitou změnu prostředí dokáží tyto ztráty minimalizovat. Opakování adaptace je v podstatě akumulace zkušeností, kterou organismus vyhodnocuje tak, že postupně minimalizuje ztráty vynaložené na adaptaci. Tento jev nazýváme *učením*.

Mezi systémy, které jsou schopny adaptace, můžeme zařadit vedle systémů přírodních samozřejmě i technické, které nás zde zajímají nejvíce. Jedná se o systémy velmi různorodé, pro jejichž popis se používá nejrůznějších matematických prostředků, a proto není lehké při definici adaptivních systémů nalézt jednotný matematický aparát. Pro potřeby definice adaptivních systémů pro řízení se omezíme na technické systémy, které splňují následující předpoklady:

- mohou měnit svůj stav nebo strukturu,
- můžeme ovlivňovat stav nebo výstup systému.

### 3.2 Adaptivní systémy

Existuje spousta definic pro pojem adaptivního systému. Jedna z jeho možných obecných definic je např.(viz. [8]):

Adaptivní systém je systém se třemi vstupy a jedním výstupem. Na adaptivní systém působí prostředí tvořené dvěma složkami: řídicí veličinou  $u$  a poruchou  $v$ . Porucha  $v$  bývá obvykle neměřitelná. Dále na systém přichází informace o jeho požadovaném chování  $\Omega$ , výstupem systému je chování systému

$$y = f(u, v, \Omega), \quad (11)$$

které přiřazuje každému projevu prostředí  $u$  a  $v$  jediný výstup  $y$ . Změna chování se uskutečňuje změnou parametrů  $\Theta$ . Za parametr  $\Theta$  se volí pro každou kombinaci  $(u, v, \Omega)$  takový parametr  $\Theta^*$ , aby minimalizoval ztrátu  $Q$ :

$$Q(\Omega, u, v, \Theta^*) = \min(\Omega, u, v, \Theta) \quad (12)$$

Adaptace je tedy proces, kterým se hledá  $\Theta^*$  a trvá tak dlouho, dokud tento parametr není nalezen. Charakteristickým rysem adaptivního systému je skutečnost,

že k procesu adaptace dochází vždy, pokud dojde ke změně projevu prostředí nebo ke změně požadovaného chování.

Pod pojmem učící se systém se rozumí systém, který po ukončení adaptace pro danou  $m$ -tou trojici  $(u_m, v_m, \Omega_m)$  posloupnosti  $(u_k, v_k, \Omega_k)$ , pro  $k = 1, 2, \dots, m, \dots, \infty$  si zapamatuje optimální hodnotu parametru  $\Theta^*$ , čímž si vytváří v paměti funkci  $\Theta^* = f(u, v, \Omega)$ .

Po ukončení učení pak pro každý projev prostředí  $u$  a  $v$  se přímo volí rozhodovací pravidlo výběrem příslušné hodnoty parametru z paměti bez adaptace. Lze tedy závěrem shrnout, že adaptivní systém adaptaci stále opakuje i při opětovném projevu prostředí a vyžaduje stále informaci o požadovaném chování, učící se systém zhodnocuje opakováné adaptace tak, že si pamatuje stav nalezený při adaptaci, kterou si ukládá do své paměti. Pomocí adaptivních systémů lze řešit celou řadu úloh:

- průběžnou (rekurzivní) identifikaci - tj. vytvoření matematického popisu řízeného objektu; používají se samočinně přizpůsobující se modely,
- Řízení systému, o nichž máme málo apriorních informací a nedovedeme proto předem stanovit strukturu a parametry řídicích algoritmů, případně takových systémů, jejichž přenosové vlastnosti se v průběhu řízení mění,
- rozpoznávání předmětů nebo situací (scén) a jejich trídění (tzv. klasifikace),
- v robotice, kde je cílem se naučit samostatně se orientovat v neznámém prostředí.

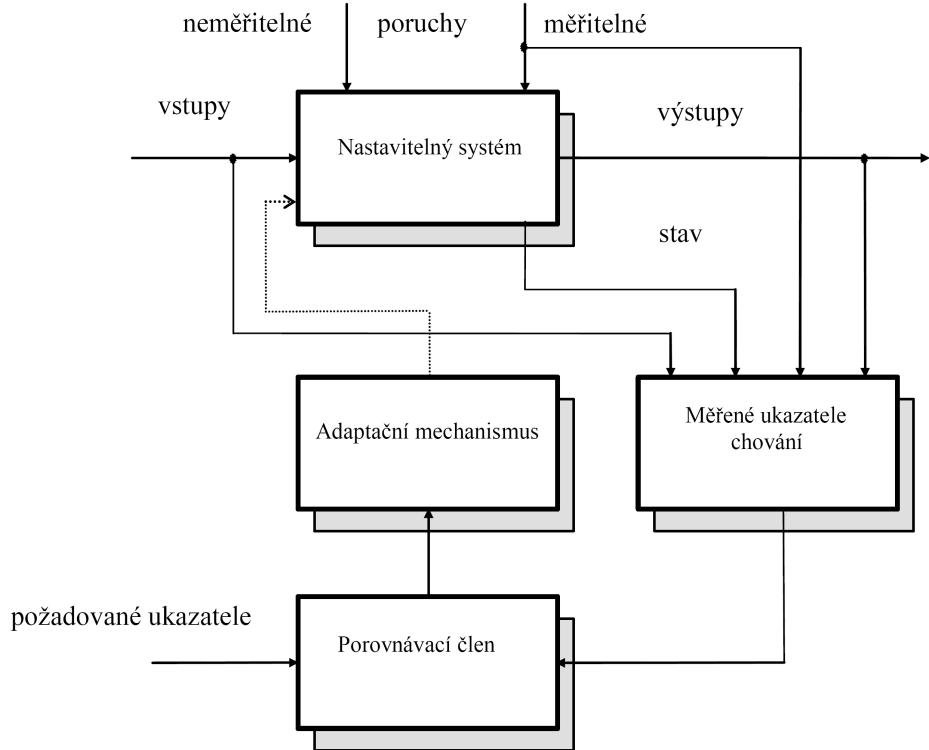
Adaptivní systém měří určité ukazatele chování daného nastavitelného systému pomocí jeho vstupů, stavů nebo výstupů. Na základě porovnání těchto měřených ukazatelů a množiny požadovaných ukazatelů modifikuje parametry nebo strukturu nastavitelného obvodu nebo generuje pomocný vstup tak, aby měřené ukazatele chování se udržovaly na hodnotách co nejbližších k žádaným ukazatelům.

Definice je dosti obecná a dovoluje zahrnout převážnou většinu adaptivních úloh technické kybernetiky. Ukazatele chování mohou být v těchto úlohách nejrůznějšího charakteru. Bude-li adaptivní systém využit pro řízení, může být ukazatelem chování např.:

- poloha pólů a nul přenosu uzavřeného regulačního obvodu v případě lineárního systému,
- žádaný překmit přechodové charakteristiky uzavřeného regulačního obvodu na skokovou změnu řídicí nebo poruchové veličiny,
- doba regulace,
- minimální hodnota různých integrálních (v diskrétní verzi sumačních) kritérií,
- amplituda a frekvence vlastních kmitů u nelineárních obvodů,
- zvolená hodnota amplitudové a fázové bezpečnosti.

### 3.3 Adaptivní řízení

Adaptivní řídící systémy přizpůsobují parametry nebo strukturu jedné části systému (regulátoru) změnám parametrů nebo struktury jiné části systému (regulované soustavy) tak, aby celý systém měl neustále optimální (v ideálním případě) chování podle zvoleného kritéria nezávisle na změnách, které nastaly.



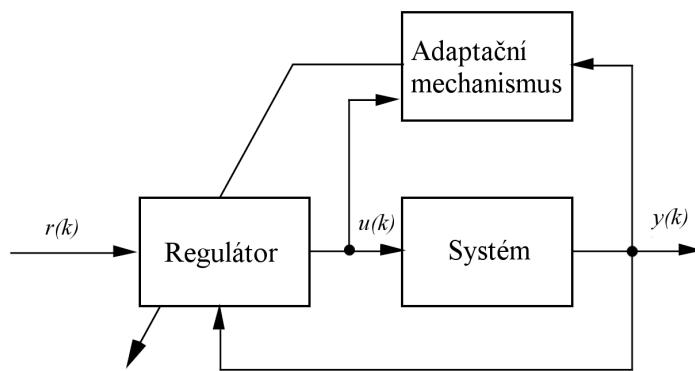
Obrázek 3: Obecné schéma adaptivního řídicího systému  
(obrázek převzat z [8])

Adaptaci na změnu parametrů nebo struktury soustavy lze uskutečnit v podstatě třemi způsoby, které jsou navzájem kombinovatelné:

- vhodnou změnou stavitelných parametrů regulátoru,
- změnou struktury regulátoru,
- generováním vhodného přídavného vstupního signálu.

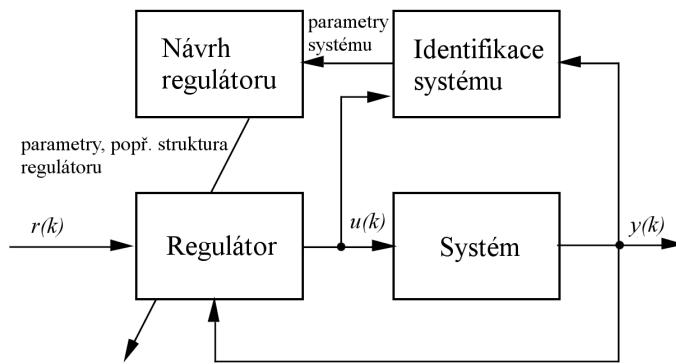
Rozdíl mezi klasickým zpětnovazebním řídicím algoritmem a adaptivním řídicím algoritmem spočívá v tom, že klasický regulátor využívá principu zpětné vazby k tomu, aby kompenzoval neznámé poruchy a stavy v procesu a sledoval změny žádané hodnoty. Zpětná vazba určuje velikost regulační odchylky  $e = r - y$ . Regulační odchylka vstupuje do pevně nastaveného řídicího algoritmu, který vypočítá hodnotu vstupního signálu (akční veličiny)  $u$  do systému (soustavy). V každé situaci je způsob zpracování regulační odchylky stejný. Podstatou adaptivního řízení je v tom, že mění způsob zpracování regulační odchylky, tj. adaptuje řídící zákon na neznámé podmínky a rozšiřuje oblast praktických případů, ve kterých lze dosáhnout kvalitní regulace. Adaptivitu lze chápat jako zpětnou vazbu vyšší úrovně, která mění parametry regulátoru podle požadované kvality regulačního pochodu.

V principu však adaptivní řídící systémy můžeme rozdělit na dvě velké oblasti. První z nich je zobrazen na obr. 4. V tomto případě adaptační algoritmus „pozoruje“ signály z řídicího systému a přizpůsobuje parametry regulátoru tak, aby byly dosaženy co možná nejlepší výsledky řízení a to i v případě, že se soustava změní. Občas jsou požadované vlastnosti charakterizovány tzv. *referenčním modelem* a adaptační algoritmus se snaží nastavit regulátor tak, aby se výsledný zpětnovazební systém choval na výstupu jako výstup z referenčního modelu. Tento přístup adaptivního řízení se nazývá „adaptivní řízení s referenčním modelem“ (angl. výraz - *model reference adaptive control (MRAC)*). V této práci použijeme jeho obdobu ve fuzzy řízení, tedy *adaptivní fuzzy řízení s referenčním modelem* (angl. výraz *model reference fuzzy control*), kde regulátor, který je laděn, je právě fuzzy regulátor.



Obrázek 4: Přímé adaptivní řízení

V druhém obecném přístupu k adaptivnímu řízení, který je zobrazen na obr. 5 se používá on-line identifikační metoda systému k odhadování parametrů soustavy, kdy na základě těchto odhadnutých parametrů systému se nastavuje regulátor pro zjištěnou soustavu. Identifikační algoritmus poskytne odhad těchto parametrů a podle některého zvoleného postupu se navrhne regulátor (v obr.5 je to blok Návrh regulátoru). V okamžiku, kdy tato metoda dokáže specifikovat regulátor pro každou množinu parametrů soustavy, bude úspěšně řídit soustavu. Tento přístup se nazývá *nepřímé adaptivní řízení*, protože ladíme regulátor nepřímo při prvním odhadu parametrů soustavy. Je to vlastně opak přímého adaptivního řízení, kde parametry regulátoru jsou odhadovány přímo bez identifikování parametrů soustavy. Tento přístup nebude v této práci uvažován.



Obrázek 5: Nepřímé adaptivní řízení

## 4 Samoučící se regulátory založené na teorii adaptivního fuzzy řízení

Bez ohledu na to, jaký je přístup návrhu pevného fuzzy regulátoru (tj. na základě znalostí experta nebo ze vstupně - výstupních dat), tak se vyskytuje jisté problémy, které ztěžují jeho návrh. Je to především:

1. vybrání vhodných funkcí příslušnosti a báze pravidel
2. dobře navržený fuzzy regulátor pro nominální soustavu nemusí fungovat adekvátně, když se parametry řízené soustavy změní nebo v případě výskytu šumu, poruch nebo jiných účinků prostředí.

Abychom byli schopni zajistit stálé dobré vlastnosti regulace v případě, že se soustava mění, musíme regulátor stále pozměňovat. V některých případech bychom mohli pro změněnou soustavu znova nastavit regulátor, ale to vždy nemůžeme. Brání tomu i spousta dalších důvodů. Je to zdlouhavé a tudíž neekonomické, mnohdy se nám to ani nemusí podařit, nehledě k tomu, že se parametry systému mění spojitě, takže by bylo potřeba teoreticky vytvořit nekonečně mnoho regulátorů, což je prakticky samozřejmě nesmysl. Bylo by tedy příjemné mít metodu, která by uměla automaticky vykonat celý návrh fuzzy regulátoru pro nominální soustavu a přitom parametry (popř. strukturu) regulátoru upravovala tak, jak by se soustava měnila.

Adaptivnímu fuzzy řízení, které je založeno na znalosti matematického popisu řízeného systému, je věnováno spousta kapitol v publikacích věnujících se fuzzy řízení. Asi nejrozšířenější je kniha [18], která navazuje na [21] a kterou doplňuje. Jiný přístup těchto metod lze najít v poměrně nové publikaci [11], ke které jsou k dispozici i simulinkové soubory k návrhu adaptivních fuzzy regulátorů, z pochopitelných důvodu však jen jako knihovní funkce bez zdrojových kódů. Praktický způsob této metody je ukázán v knize [12]. Nepřímým adaptivním fuzzy řízením se zabývá např. kniha [14], popř. [20], která je však matematicky již velmi náročná. Další publikace, které část svého obsahu věnují této oblasti jsou např. [10], [15] nebo [17].

### 4.1 Samoorganizující se fuzzy regulátor

Poprvé byl adaptivní fuzzy regulátor představen v článku [22]. Tímto článkem se začala pozvolna rozvíjet nová rozšířená metoda návrhu fuzzy regulátoru. Druhý významný článek byl [23], který navazoval na [22] a který uvedenou metodu dále rozpracoval. Zatímco v češtině existuje málo pojmu, kterými bychom mohli takový regulátor pojmenovat, tak v angličtině existuje celá řada pojmu, např. adaptive, self-tuning, self-structuring, self-regulating, self-organizing nebo self-learning. Je poměrně obtížné určit hranice mezi těmito pojmy. Poměrně detailně se těmito rozdíly zabývá část publikace [17]. Různí autoři ale stejně používají tyto pojmy a jejich významy celkem benevolentně.

V době vydání článku [22] nebyly ještě zřetelné rozdíly mezi pojmy adaptivní a samoorganizující. Dnes se hodně často bere pojem adaptivní jako nadřazený, do kterého patří všechny ostatní výše uvedené pojmy. Výrazem *samořídící* (self-tuning) se míní ladění pouze zesílení regulátoru. *Samoorganizující* (self-organizing, self-structuring, self-regulating) regulátor navíc mění i strukturu regulátoru. *Samoučící se* (self-learning) regulátor navíc pracuje s pamětí, proto má schopnost si akumulovat získané informace,

které pak využije. Nad tím vším může stát ještě tzv. *supervizor regulátor*, který jednotlivé regulátory pod sebou řídí.

Mamdani se svými studenty jako první vyvinuli samoorganizující se regulátor jako rozšíření klasického fuzzy PD regulátoru. Nazvali ho právě samoorganizující se (dále jen SOC), protože byl schopený přizpůsobovat bázi znalostí fuzzy regulátoru bez lidského zásahu. Tento regulátor byl navržen především pro řízení systémů s jedním vstupem a jedním výstupem a především pro systémy s dopravním zpožděním.

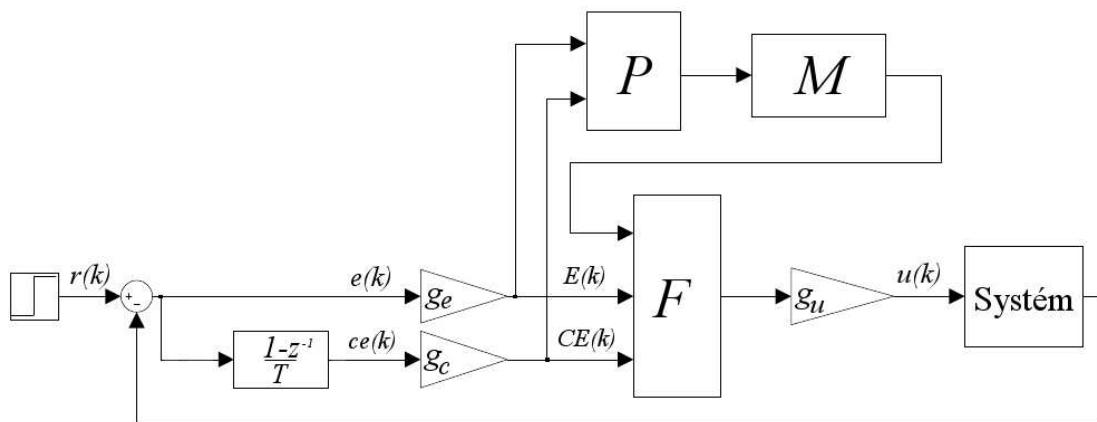
I zde platí podobné principy jako pro klasické adaptivní řízení, proto se regulační obvod skládá ze dvou smyček:

1. vnitřní, která je klasická zpětnovazební smyčka
2. vnější, která ladí parametry nebo strukturu regulátoru

Na obr. 6 je vidět samoorganizující se fuzzy PD regulátor ve zpětnovazebním zapojení. Tento regulátor se kromě zesílení  $g_e$ ,  $g_c$  a  $g_u$  skládá ještě z bloků  $F$ ,  $P$  a  $M$ . Blok  $F$  spolu se třemi zmíněnými zesíleními tvoří dohromady klasický fuzzy PD regulátor.

V bloku  $F$  jsou obsažena pravidla řízení, která budou modifikována. Na vstupu pro odchylku  $E(k)$  a na vstupu pro diferenční odchylku  $CE(k)$  je použito 13 trojúhelníkových funkcí příslušnosti, které jsou rovnoměrně rozmištěny. Vzhledem k tomu, že je použito velké množství funkcí příslušnosti, nepoužívají se zde pro jednotlivé funkce příslušnosti slovní označení, ale pouze číselná označení, tj. funkce příslušnosti umístěná nejvíce vlevo má při celkovém počtu 13 funkcí na jednom vstupu identifikátor -6 (funkce nejvíce vpravo +6). V tomto případě tvoří bázi pravidel dohromady  $13 * 13 = 169$  pravidel. Vzhledem k tomu, že se jedná o velké množství pravidel, tak jsou kvůli přehlednosti pravidla zobrazována v tabulce. Řádky tvoří jednotlivé odchylky  $E$  a sloupce jednotlivé diference  $CE$ , hodnoty v tabulce jsou čísla představující již zmíněné číselné identifikátory jednotlivých výstupních trojúhelníkových funkcí, které zároveň představují středy jednotlivých výstupních funkcí.

Blok  $P$  představuje tzv. *specifikaci kvality regulace*, která určuje, jak se v jednotlivých stavech systému bude fuzzy PD regulátor, přesněji pravidla obsažená v bloku  $F$ , modifikovat. Blok  $M$  má pak na starost tuto modifikaci provést. Modifikace spočívá v posunutí středu výstupní funkce a tedy ve změně akčního zásahu.



Obrázek 6: Samoorganizující se fuzzy PD regulátor ve zpětnovazebním zapojení

## Podrobněji vysvětlené jednotlivé části SOC

Při návrhu SOC nemusíme znát matematický popis systému, ani jeho strukturu. Požaduje se jen, aby statická charakteristika systému byla monotónní. Systém může být nelineární a může obsahovat velké dopravní zpoždění, které však zůstává po celou dobu regulace konstantní. Dominantní časová konstanta a dopravní zpoždění stačí být známy pouze přibližně. Obě přibližné hodnoty můžeme získat z přechodové charakteristiky.

Blok  $P$  má na starost zvolit, na základě stavu systému, vhodnou penalizaci, aby se dosáhlo požadovaného průběhu na výstupu. Kvalita regulace se určuje na základě aktuální hodnoty odchylky  $E(k)$  od požadované hodnoty a její diference  $CE(k)$ . Specifikace určení kvality představuje opět fuzzy PD regulátor, který však na výstup v tomto případě dává  $p(k)$ , tedy velikost posunutí výstupní funkce příslušnosti. Tento fuzzy PD regulátor má také 13 funkcí příslušnosti na obou vstupech, stejně jako fuzzy PD regulátor v bloku  $F$ . Vzhledem k tomu, že opět vychází velké množství pravidel, nepoužívá se ani tady slovní označení pro jednotlivé funkce příslušnosti, ale pouze číselné označení a z důvodu přehlednosti se pravidla zobrazují opět v tabulce.

Příkladem takových pravidel je Tabulka 2 a Tabulka 3. Jednotlivé nuly v obou tabulkách představují požadovaný stav systému bez nutnosti penalizace. V Tabulce 2 tvoří nuly tvar písmene  $Z$ , které určují, že požadovaným stavem může být i překmit žádané hodnoty. V Tabulce 3 jsou nuly pouze na diagonále, proto se penalizují i stavy, kdy dochází k překmitu. Z tohoto důvodu Tabulka 2 představuje požadovanou odezvu systému s překmitem a Tabulka 3 odezvu systému bez překmitu.

Jinými slovy, blok  $P$  vyjadřuje, jaká kombinace  $E(k)$  a  $CE(k)$  je žáданá pro požadovaný průběh a jaká ne, tím vlastně určují požadovaný výstup systému. Tato tabulka může být vygenerovaná použitím pravidel, podobně jako navrhujeme klasický fuzzy regulátor. Častěji je však vygenerovaná ručně.

Regulátor může začít pracovat i s nulovou bází pravidel, ale rychleji dokonverguje k ustálenému řešení, když bude „nastartován“ se smysluplnými hodnotami.

Základní myšlenkou SOC je nastavit přizpůsobovací mechanismus, tedy blok  $P$ . V případě, že je regulace nedostatečná, bude pravidlo ve fuzzy regulátoru (blok  $F$ ), která způsobila špatný stav, modifikováno. Tím se dosáhne toho, že když se systém znovu dostane do stejného stavu, bude použito modifikované pravidlo, které přiblíží výstup požadované hodnotě. Nemusí to však platit obecně, záleží na návrhu přizpůsobovacího algoritmu, který je vždy navržen na základě heuristických znalostí.

Pro výpočet akčního zásahu  $u(k)$  a výpočet penalizace  $p(k)$  byl použit inferenční mechanismus Max - Min a následně defuzzifikační metoda těžiště.

### Jak samotný adaptační algoritmus pracuje

SOC se adaptuje na základě požadované odezvy, kterou chceme mít na výstupu. V časový okamžik  $k$  SOC

- získává informace o tom, jak je daleko od požadovaného stavu,
- a na základě toho modifikuje pravidla v bloku  $F$ .

Blok  $P$  (resp. tabulka  $P$ ) na základě zjištěného stavu vrátí hodnotu  $P(i_k, j_k)$ . Index  $i_k$  koresponduje k  $E(k)$  tak, že  $E(k) = U_e(i_k)$ , kde  $U_e$  je vstupní rozsah. Index  $j_k$  koresponduje k  $CE(k)$  tak, že  $CE(k) = U_{ce}(j_k)$ , kde  $U_{ce}$  je druhý vstupní rozsah.

$P(i_k, j_k)=0$  znamená, že stav  $(E(k), CE(k))$  je splněn. V případě, že  $P(i_k, j_k)$  je nenulové, pak je stav, ve kterém se právě nacházíme, neuspokojivý. Když  $p(k)$  je

nenulové, tak modifikační algoritmus  $M$  posune výstupní funkci o  $p(k)$ . Toto pravidlo však ve skutečnosti nemusí být za špatný stav zodpovědné, protože to mohlo způsobit pravidlo v nějakém čase o  $d$  kroků předtím.

Řešení je vrátit se o tento počet kroků  $d$  do minulosti a tam toto pravidlo korigovat. Je potřeba toto  $d$  určit. To souvisí s dopravním zpožděním systému a také s časovou konstantou systému. Proměnná  $d$  se v angličtině označuje jako *delay in penalty* nebo *delay in reward*.

Protože předpokládáme, že máme monotónní statickou charakteristiku systému, tak můžeme říci, že akční zásah  $u(k - d)$  měl být změněn na  $u(k - d) + p(k)$ , což můžeme zapsat jako

$$u(k - d) = u(k - d) + p(k). \quad (13)$$

A vzhledem k tomu, že pravidla jsou vyjádřena ve formě tabulky, můžeme přepsat předchozí vztah na následující:

$$F(i, j)_{k-d} = F(i, j)_{k-d} + P(i, j)_k \quad (14)$$

Pro připomenutí uvedeme, že časový index  $k$  určuje aktuální časový okamžik, index  $k - d$  určuje časový okamžik o  $d$  časových kroků dříve.

| Proměnná | Časový okamžik |       |       |       |       |
|----------|----------------|-------|-------|-------|-------|
|          | $k=1$          | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
| $E$      | 6              | 3     | 1     | 0     | -1    |
| $CE$     | 0              | -3    | -2    | -1    | -1    |
| $u$      |                | 0     | -1    | -1    | -2    |
| $p$      |                | 0     | 0     | 0     | -1    |

Tabulka 1: Ukázka výpočtu penalizace

Předpokládejme  $d = 2$  a proměnné v Tabulce 1 po jednotkovém skoku jako referenci. Od času  $t = 1$  do  $t = 4$ , se výstup soustavy pohybuje směrem k referenčnímu signálu. Zřejmě sleduje referenci, protože specifikace kvality dává na výstupu  $p = 0$ . V čase  $t = 5$  chyba  $E$  změní znaménko, které indikuje překmit, a tabulka specifikace kvality na to reaguje tak, že  $p(5) = -1$ . Protože  $d = 2$ , tak modifikace v tabulce  $F$  bude na pozici odpovídající času  $t = n - d = 5 - 2 = 3$ . V tomto čase byl stav  $(E(3), CE(3)) = (1, -2)$  a element  $F(i, j) = -1$ . Modifikovaná položka je z tohoto důvodu  $u(3) = u(3) + p(5) = -1 - 1 = -2$ , což bude vloženo do  $F(i, j)_3$ .

Původní pravidla specifikace kvality regulace (tedy Tabulka 2 nebo Tabulka 3) byly vytvořeny ručně metodou pokus omyl. Dejme tomu, že čísla v tabulce  $P$  jsou malá, pak je potřeba mnoho modifikací než  $F$  dokonverguje do ustáleného stavu. Na druhé straně, pokud jsou čísla v tabulce  $P$  v absolutní hodnotě velká, konvergence by měla být rychlejší, ale může se stát, že se systém stane nestabilní. Musíme proto volit kompromis.

| E  | CE |    |    |    |    |    |    |    |    |    |   |   |   |   |
|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|
|    | -6 | -5 | -4 | -3 | -2 | -1 | 0  | 1  | 2  | 3  | 4 | 5 | 6 |   |
| -6 | -6 | -6 | -6 | -6 | -6 | -6 | -6 | 0  | 0  | 0  | 0 | 0 | 0 | 0 |
| -5 | -6 | -6 | -6 | -6 | -6 | -6 | -6 | -3 | -2 | -2 | 0 | 0 | 0 | 0 |
| -4 | -6 | -6 | -6 | -6 | -6 | -6 | -6 | -5 | -4 | -2 | 0 | 0 | 0 | 0 |
| -3 | -6 | -5 | -5 | -4 | -4 | -4 | -4 | -3 | -2 | 0  | 0 | 0 | 0 | 0 |
| -2 | -6 | -5 | -4 | -3 | -2 | -2 | -2 | 0  | 0  | 0  | 0 | 0 | 0 | 0 |
| -1 | -5 | -4 | -3 | -2 | -1 | -1 | -1 | 0  | 0  | 0  | 0 | 0 | 0 | 0 |
| 0  | -4 | -3 | -2 | -1 | 0  | 0  | 0  | 0  | 0  | 1  | 2 | 3 | 4 | 4 |
| 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 2  | 3 | 4 | 5 | 5 |
| 2  | 0  | 0  | 0  | 0  | 0  | 0  | 2  | 2  | 2  | 3  | 4 | 5 | 6 | 6 |
| 3  | 0  | 0  | 0  | 0  | 2  | 3  | 4  | 4  | 4  | 4  | 5 | 5 | 6 | 6 |
| 4  | 0  | 0  | 0  | 2  | 4  | 5  | 6  | 6  | 6  | 6  | 6 | 6 | 6 | 6 |
| 5  | 0  | 0  | 0  | 2  | 2  | 3  | 6  | 6  | 6  | 6  | 6 | 6 | 6 | 6 |
| 6  | 0  | 0  | 0  | 0  | 0  | 0  | 6  | 6  | 6  | 6  | 6 | 6 | 6 | 6 |

Tabulka 2: Specifikace kvality regulace (blok  $P$ ) - Procyk,  
Mamdani 1979 (převzato z [15])

| E  | CE |    |    |    |    |    |    |    |    |    |    |    |   |   |
|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|
|    | -6 | -5 | -4 | -3 | -2 | -1 | 0  | 1  | 2  | 3  | 4  | 5  | 6 |   |
| -6 | -6 | -6 | -6 | -6 | -6 | -6 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 0 |
| -5 | -6 | -6 | -6 | -6 | -5 | -4 | -4 | -4 | -3 | -2 | -1 | 0  | 0 | 0 |
| -4 | -6 | -6 | -6 | -5 | -4 | -3 | -3 | -3 | -2 | -1 | 0  | 0  | 1 | 1 |
| -3 | -6 | -6 | -5 | -4 | -3 | -2 | -2 | -2 | -1 | 0  | 0  | 1  | 2 | 2 |
| -2 | -6 | -5 | -4 | -3 | -2 | -1 | -1 | -1 | 0  | 0  | 1  | 2  | 3 | 3 |
| -1 | -5 | -4 | -3 | -2 | -1 | -1 | 0  | 0  | 0  | 1  | 2  | 3  | 4 | 4 |
| 0  | -5 | -4 | -3 | -2 | -1 | 0  | 0  | 0  | 1  | 2  | 3  | 4  | 5 | 5 |
| 1  | -3 | -2 | -1 | 0  | 0  | 0  | 0  | 1  | 1  | 2  | 3  | 4  | 5 | 5 |
| 2  | -2 | -1 | 0  | 0  | 0  | 1  | 1  | 1  | 2  | 3  | 4  | 5  | 6 | 6 |
| 3  | -1 | 0  | 0  | 0  | 1  | 2  | 2  | 2  | 3  | 4  | 5  | 6  | 6 | 6 |
| 4  | 0  | 0  | 0  | 1  | 2  | 3  | 3  | 3  | 4  | 5  | 6  | 6  | 6 | 6 |
| 5  | 0  | 0  | 1  | 2  | 3  | 4  | 4  | 4  | 5  | 6  | 6  | 6  | 6 | 6 |
| 6  | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 6  | 6  | 6  | 6  | 6  | 6 | 6 |

Tabulka 3: Specifikaci kvality regulace (blok  $P$ ) - Yamazaki 1982  
(převzato z [15])

### Ladění parametrů SOC

Připomeňme, že laděním (angl. tuning) máme na mysli především vhodné změny zesílení regulátoru, zatímco pojmem samoorganizující se míní navíc i změna báze znalostí.

- **Zesílení**  $g_e$ ,  $g_c$ ,  $g_u$

Zesílení regulátoru musí být nastaveno kolem smysluplných hodnot, přesná volba ale zde není opět až tak důležitá jako u pevného fuzzy regulátoru. Představme si např., že zvolíme výstupní zesílení malé. Adaptační algoritmus se bude snažit vykompenzovat nedostatečnou regulaci tím, že bude posouvat středy výstupních funkcí směrem od středu, čímž bude stejně zvyšovat potřebný akční zásah a tím kompenzovat nízko zvolené výstupní zesílení. Jako vhodný způsob se jeví

navrhnut zesílení podle pravidel nastavení klasického PD regulátoru. Toto se zdá být vhodný způsob pro samoorganizující se fuzzy regulátor.

- **Zpoždění  $d$**

Parametr  $d$  měřený v časových krocích by měl být vybíráno s ohledem na periodu vzorkování  $T$ . Heuristický výběr se volí zpravidla podle následujícího vztahu:

$$d = \text{round}(\tau/T),$$

kde  $\tau$  je časová konstanta systému.

- **Adaptační zesílení  $\gamma$**

Čím větší zesílení  $\gamma$ , tím rychleji čísla v tabulce  $F$  rostou, ale když je tento parametr nastaven příliš vysoko, adaptace se stavá nestabilní. Jako rozumná volba se jeví výběr podle následujícího vztahu:

$$\gamma \leq \frac{0.2 * |F(i,j)|_{\max}}{|E(k) + \tau * CE(k)|_{\max} * T}$$

#### 4.1.1 Použití SOC na konkrétním systému

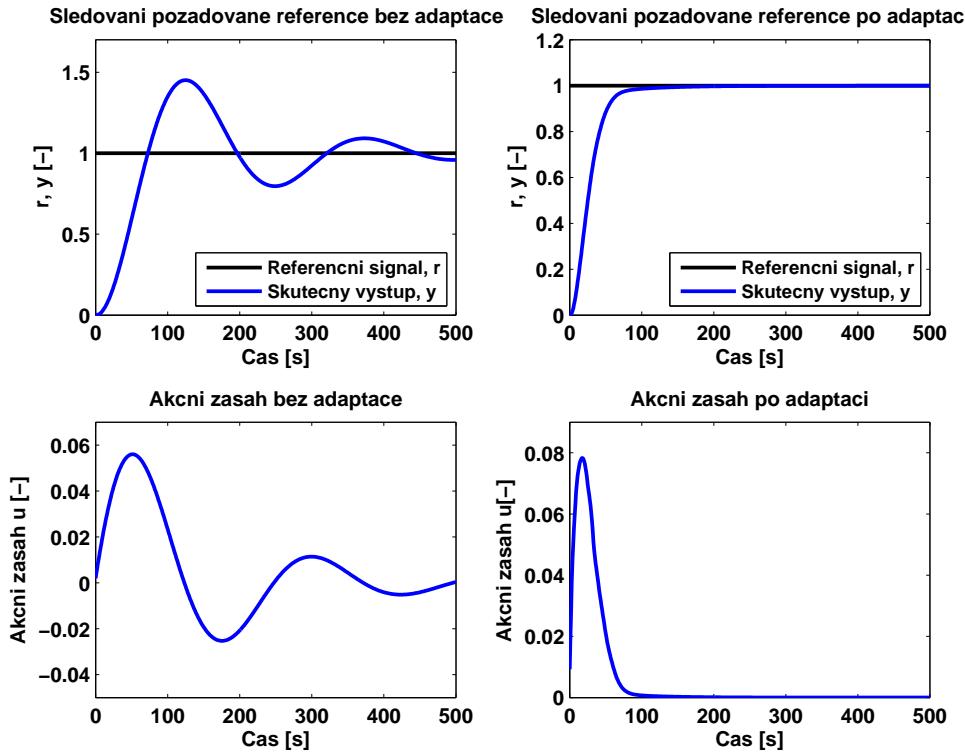
Pro následující simulace byl zvolen systém s uvedeným přenosem:

$$P = \frac{1}{s(s+2)(s+1.5)} \quad (15)$$

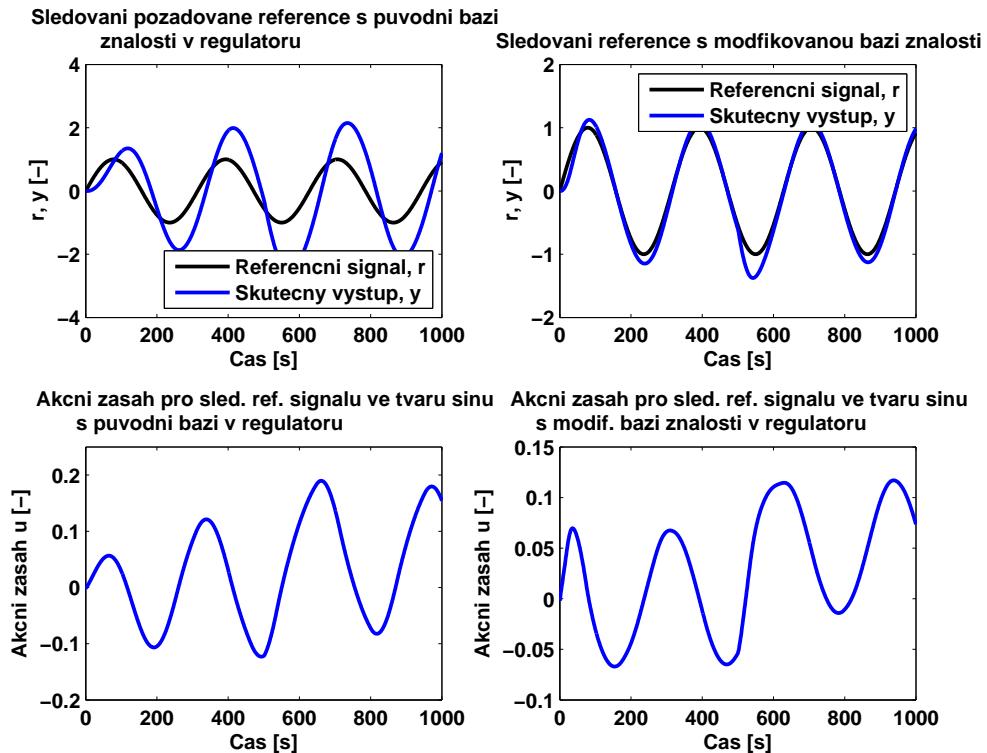
Startovní báze pravidel v regulátoru je zobrazena na obr.9 vlevo. Jedná se o klasický lineární PD regulátor. Na následujícím obr.7 jsou zobrazeny průběhy sledování jednotkového skoku při vypnuté adaptaci (horní vlevo) a při zapnuté adaptaci (horní vpravo). Z těchto obrázků je vidět, že oba regulátory dokážou na uvedený jednotkový skok regulovat. Bez adaptace však dochází k velkému překmitu (až 50%). Výsledná báze pravidel po adaptaci je zobrazena na obr.9 vpravo. Akční zásahy vztázené k jednotlivým regulacím jsou uvedené pod těmito průběhy. Z nich je vidět, že navíc akční zásah použitý při adaptaci má mnohem menší průměrnou hodnotu. Z uvedeného vyplývá, že v případě jednotkového skoku vychází jednoznačně lépe regulace s adaptací.

Velmi často však chceme regulovat na jiné žádané průběhy než jen na jednotkový skok. Pokud chceme použít adaptaci, tak se však dostaneme do potíží, jak navrhnut fuzzy regulátor v bloku  $P$ , který by měl volit vhodnou penalizaci v určitých okamžicích. U obecného průběhu je totiž témeř nemožné vytvořit analogická pravidla pro penalizaci jako v případě jednotkového skoku, protože těžko můžeme u takového průběhu určit, co je doba náběhu, překmit nebo třeba doba ustálení. Jediná možnost se tudíž nabízí nechat regulátor naadaptovat na jednotkový skok a následnou bázi pravidel použít pro jiný požadovaný průběh. Tento případ zobrazuje obr.8.

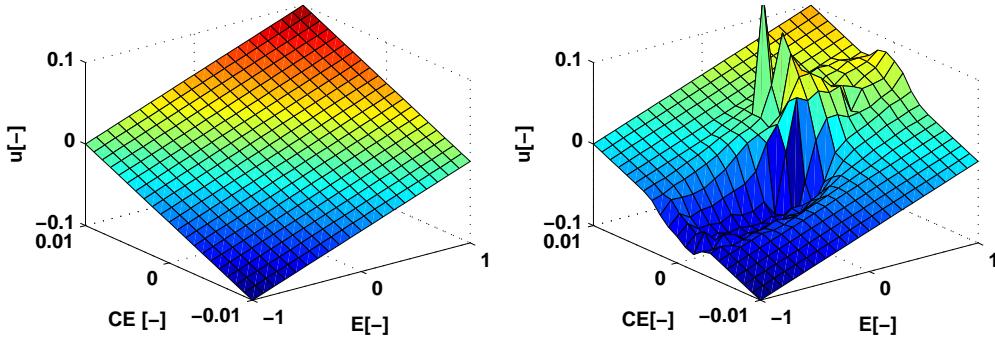
Na zmíněném obrázku jsou opět zobrazeny průběhy sledování požadovaného průběhu při použití lineární báze pravidel (obr.9 vlevo) a vypnuté adaptaci (obr.8 nahoře vlevo) a při modifikované bázi pravidel (obr.9 vpravo), která vznikla adaptací na jednotkový skok (obr.8 nahoře vpravo). Z těchto průběhů je vidět, že sice došlo k výraznému zlepšení, ale i tak to není ještě dostatečná regulace. Akční zásah v případě použité modifikované báze pravidel vychází v průměru o trochu menší než v případě použití nemodifikované báze pravidel.



Obrázek 7: Průběhy při sledování jednotkového skoku při vypnuté adaptaci (obrázky vlevo) a při zapnuté adaptaci (obrázky vpravo)



Obrázek 8: Sledování sinusového průběhu při použití nemodifikované tabulky (obrázky vlevo) a při použití modifikované tabulky, která vznikla z adaptace na jednotkový skok (obrázky vpravo) z předchozího obr. 7



Obrázek 9: Nemodifikovaná (=startovní) (obr. vlevo) a modifikovaná báze po procesu adaptace (obr. vpravo)

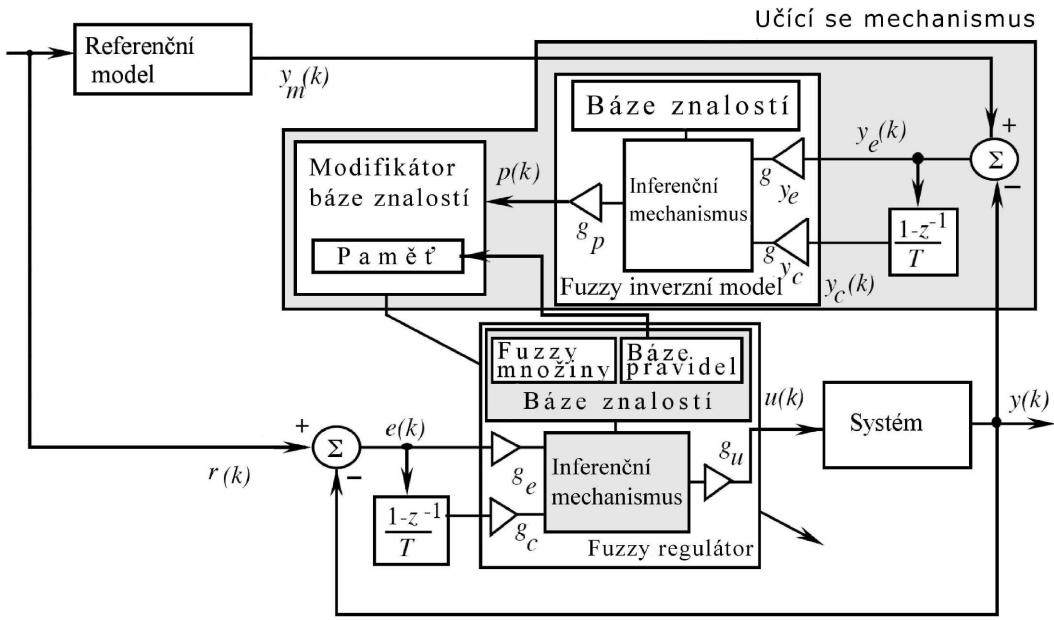
## 4.2 Adaptivní fuzzy regulátor na principu učení s referenčním modelem

Velkým nedostatkem předchozí metody je obtížnost vytvoření penalizační tabulky  $P$  pro obecný požadovaný průběh. Sami autoři článku [22] píší, že pro dané systémy byl vytvořen metodou *pokus - omyl*, k tomu různé systémy potřebují různou penalizační tabulkou pro dosažení stejného požadovaného průběhu. Navíc v případě systémů s více stavami může být pouze PD regulátor nedostatečný. Z tohoto důvodu musíme přidat další vstupy, které však situaci ještě více zkomplikují. Výše zmíněné nedostatky vedou k tomu, že uvedený postup je silně omezený pouze pro určitou skupinu jednoduchých systémů. Z tohoto důvodu byl vytvořen pokročilejší algoritmus, který je popsán v této kapitole.

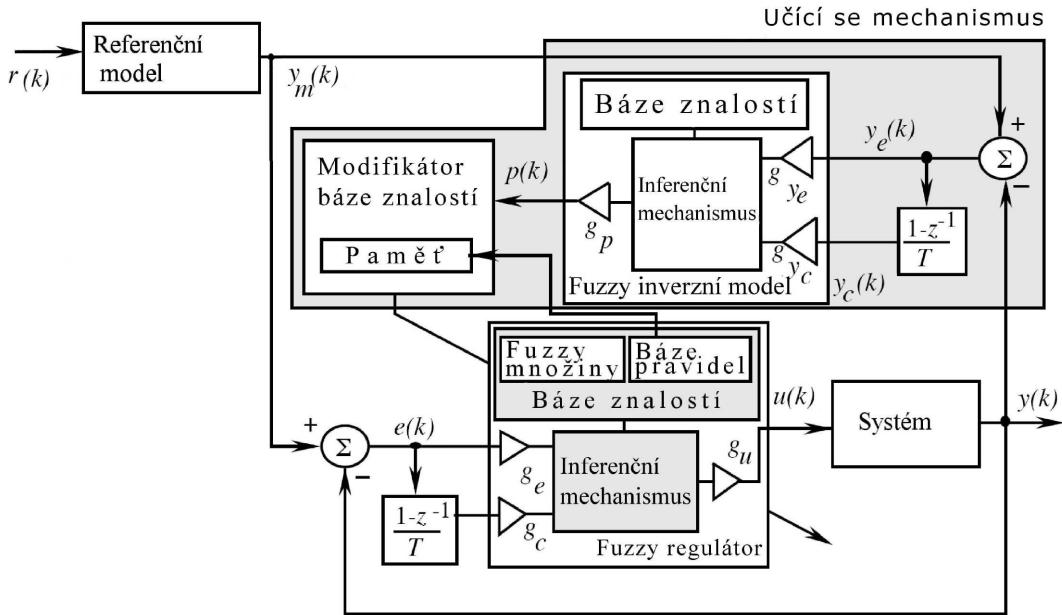
Celá tato sekce je proto věnována tzv. *adaptivnímu fuzzy regulátoru na principu učení s referenčním modelem* (v lit. tuto metodu najdeme pod pojmem *fuzzy model reference learning control - FMRLC*), který je vlastně modifikací přímého adaptivního regulátoru (viz. obr.4). Algoritmus je popsán především v publikacích [16], [13] a částečně v [19]. Z českých zdrojů je to prakticky jediná práce [9].

Výraz *učení* je zde použit za účelem zdůraznění rozdílu mezi tímto pojmem a pojmem *adaptivní*. Tento rozdíl byl již diskutován v sekci 3.2. Mnoho konvenčních adaptivních metod řízení pro lineární systémy se stále přeladují v jednotlivých pracovních bodech, což je nevhodné, a proto se tomuto nedostatku tato metoda snaží zabránit. Navíc význam FMRLC spočívá především v tom, že je použitelný i pro nelineární systémy. V souladu s obecnou definicí (12) adaptivního systému tu  $u$  představuje akční zásah do systému, požadované chování  $\Omega$  tu bude představováno požadovaným výstupním průběhem a parametr  $\Theta$ , kterým se ukutečňuje změna chování, ten bude představován posouváním výstupních funkcí regulátoru (podrobněji to bude vysvětleno dále), poruchy  $v$  tu nepředpokládáme. Učící se systém navíc bude znamenat, že si pro jednotlivé parametry  $u$ ,  $v$  (v našem případě stále nula) a  $\Omega$  zapamatuje optimální hodnotu parametru  $\Theta^*$ , který představuje naučenou bázi pravidel, kterou uložíme do paměti.

Schéma FMRLC je na obr.10. Má čtyři hlavní části: *systém* (tj., co chceme řídit), *fuzzy regulátor*, který bude laděn, *referenční model* a *učící se mechanismus* (v případě, že nebudeme používat paměť, stává se z něho *adaptační mechanismus*).



Obrázek 10: Adaptivní fuzzy regulátor na principu učení s referenčním modelem (1)  
 (= fuzzy model reference learning controller - FMRLC, poupraveno z [16])



Obrázek 11: Adaptivní fuzzy regulátor na principu učení s referenčním modelem (2)  
 (= fuzzy model reference learning controller - FMRLC, obr. poupraven z [16])

Vzhledem k tomu, že se tu bude jednat o trochu jiný přístup než v předchozí kapitole, budeme také považovat v souladu s obr.10 vstupy do regulátoru označené jako  $e(k)$  a diferenci odchylky jako  $ce(k)$ , zatímco v předchozí kapitole jsme jako vstupy

do fuzzy regulátoru, představovaný blokem  $F$ , označovali vstupy  $E(k)$  pro odchylku  $CE(k)$  pro její diferenci. Pokud se tu budeme zmiňovat o vstupech do regulátoru, budeme tím implicitně myslet vstupy  $e(k)$  a  $ce(k)$ , to znamená ještě před vynásobením zesílením.

Učící se mechanismus pracuje na základě pozorovaných dat  $y_m(k)$  a  $y(k)$  z fuzzy řídicího systému. Jako doplňující data mohou sloužit  $r(k)$  ( $T$  je vzorkovací perioda). Tato data pomáhají charakterizovat celý zpětnovazební systém. Vlastnosti výstupních průběhů tohoto systému se snažíme mít co nejvíce podobné výstupu z referenčního modelu  $y_m(T)$ . Jedná se o analogický přístup v konvenčním adaptivním řízení s referenčním modelem. Učící se mechanismus se snaží přizpůsobovat fuzzy regulátor tak, aby se celý zpětnovazební systém (tj. zobrazení z  $r(k)$  do  $y(k)$ ) choval jako výstup z referenčního modelu.

Podle knihy [16] se klasická zpětná vazba na obr.10 snaží dosáhnout takové  $y(k)$ , aby co nejlépe sledovalo  $r(k)$ . Toho se dosahuje působením akčního zásahu  $u$  do systému. Horní část s učícím se mechanismem se naopak snaží ludit regulátor tak, aby výstup  $y(k)$  měl stejný průběh jako výstup z referenčního modelu  $y_m(k)$ . Penalizace  $p(k)$  tak mění bázi pravidel regulátoru tím, že posouvá středy výstupních trojúhelníkových funkcí příslušnosti. Penalizace  $p(k)$  je výstupem z učícího se mechanismu a „vstupem“ do regulátoru.

Zdá se tedy, že by se mohlo jednat o dva protichůdné požadavky. Z tohoto důvodu autor této práce zavedl modifikaci zapojení, které je zobrazeno na obr.11. Tím se dosáhlo toho, že jak klasická zpětnovazební smyčka, tak horní smyčka s učícím se mechanismem se snaží dosáhnout toho, aby výstup  $y(k)$  sledoval jeden společný průběh, tj.  $y_m(k)$ , který se tak vlastně stává referenčním signálem, který chceme sledovat. Uvedená změna se velmi pozitivně projevila především u řízení nelinárních systémů. K této změně zapojení se ještě vrátíme při aplikaci FMRLC na konkrétních modelech. Vysvětlení FMRLC bude ale dále vztaženo k obr.10, protože princip jinak zůstává stejný.

Jednotlivé komponenty FMRLC budou z důvodu názornosti popsány detailněji pro systém s jedním vstupem a jedním výstupem (tzv. SISO systém). Pro systémy s více vstupy a více výstupy (tzv. MIMO systémy) je postup analogický.

#### 4.2.1 Fuzzy regulátor

Systém má vstup  $u(k)$  a výstup  $y(k)$ . Připomeňme, že v této práci používáme pouze fuzzy PD regulátor, ale obecně můžeme ve FMRLC použít fuzzy regulátor i s jinými vstupy než jen s odchylkou  $e(k)$  a diferencí odchylky  $ce(k)$ . Obecně je velmi důležité správně vybírat vstupy do regulátoru, ale zároveň i rozumné reference  $r(k)$ . V opačném případě můžou být vlastnosti regulace nepříznivě ovlivněny a zároveň je nemožné zajistit stabilitu. Zamezení stále rostoucího akčního zásahu nade všechny meze je možno provést tím, že před každou adaptací ve fuzzy regulátoru nejdříve zjistíme, zda daná změna neumožní právě takový akční zásah  $u$ , který bude mimo povolený rozsah. V případě, že tato situace nastane, tak k adaptaci nedojde. Toto je v této práci při aplikaci FMRLC na konkétních modelech bráno v úvaze.

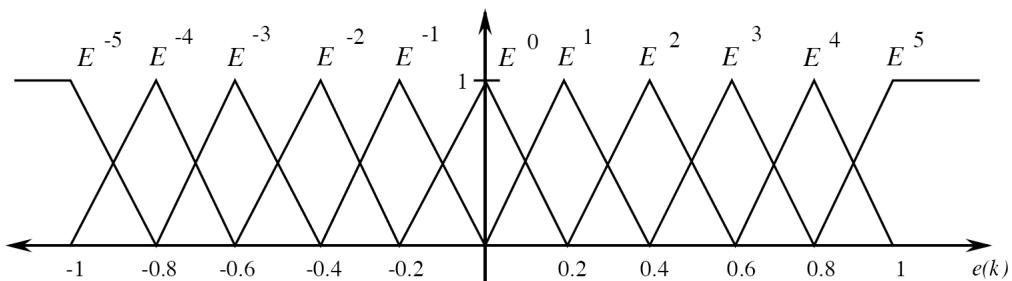
Jak je vidět v obr. 10, volíme pro odchylku  $e(k)$  zesílení  $g_e$ , pro diferenci odchylky  $ce(k)$  zesílení  $g_c$  a pro akční zásah  $u(k)$  zesílení  $g_u$ . První odhad těchto zesílení může být podle následujícího postupu.

Zesílení  $g_e$  může být vybráno tak, že rozsah hodnot  $e(k)$  bude takový, aby aktuální hodnoty  $e(k)$  nebyly mimo nejvzdálenější funkce příslušnosti, tj. nebyly v saturaci.

Zesílení  $g_c$  může být určeno experimentováním s různými vstupy  $r(k)$  s vypnutým adaptačním mechanismem. Využitím této znalosti nastavíme  $g_c$  tak, že obvyklé hodnoty nebudou v saturaci nejvzdálenějších funkcí příslušnosti. Zesílení  $g_u$  vybereme tak, že tvoří maximum povoleného akčního zásahu  $u$  do systému. Výběr těchto zesílení je však heuristický a nemusí vždy fungovat. Někdy jsou potom tato zesílení laděna společně se všemi parametry FMRLC.

### Báze pravidel

Ve FMRLC algoritmu používáme na všech vstupech trojúhelníkové funkce příslušnosti, které jsou rovnoměrně rozloženy, přičemž na každém vstupu jich je stejný počet. V případě 11 funkcí je případ zobrazen na následujícím obr. 12.



Obrázek 12: Rozložení funkcí příslušnosti na vstupu, v tomto případě na vstupu pro odchylku  $e$ , stejně funkce včetně rozložení je i pro vstup diference odchylky  $ce$

Při FMRLC budeme používat všechny možné kombinace v bázi pravidel. Pokud máme 11 funkcí příslušnosti na každém ze dvou vstupů, dostáváme  $11 * 11 = 121$  pravidel.

### Inicializace báze znalostí

Vstupní funkce příslušnosti charakterizují předpoklad (= antecedent) pravidel, určují různé situace, ve kterých budou pravidla aplikována. Vstupní funkce příslušnosti jsou neměnné během celého průběhu FMRLC algoritmu. Naopak výstupní funkce příslušnosti jsou neznámé. Ty se právě FMRLC algoritmus snaží najít a zároveň s nimi také pravidla. FMRLC algoritmus se tak snaží najít pro různé situace charakterizované předpoklady takové akční zásahy, které povedou k co nejlepší regulaci.

Musíme vybrat inicializační hodnoty pro každou z výstupních funkcí příslušnosti. Např. pro rozsah výstupu  $[-1,1]$  můžeme vybrat trojúhelníkové funkce příslušnosti se základnou o velikosti 0.4 a středy v nule. Tento výběr reprezentuje fuzzy regulátor tak, že regulátor nemá žádné informace o tom, jak systém řídit. Vstup do systému je zinicializován na  $u = 0$ . Někdy může být rozumné dodat alespoň nějaká hrubá pravidla o tom, jak systém řídit, než jednoduše nahradit všechny středy výstupních funkcí nulami. Jako defuzzifikační metodu budeme používat standardní metodu těžiště.

### Jak vybírat vstupy a jejich souvislost s adaptací a učením

Výběr vstupů do regulátoru obecně zahrnuje i problémy související s učením, resp. adaptací FMRLC. Jak FMRLC pracuje, učící se mechanismus bude ladit výstupní funkce příslušnosti fuzzy regulátoru. Především se ale bude regulátor pro každou různou kombinaci  $e(k)$  a  $ce(k)$  snažit naučit, které akční zásahy jsou pro regulaci nejlepší.

Obecně ale existuje těsná souvislost mezi tím, co vstupy poskytují regulátoru a schopnosti regulátoru učit se řídit systém, a to pro různé referenční vstupy a různé pracovní podmínky systému. FMRLC by měl být schopen se naučit a zapamatovat si různé regulátory pro všechny různé pracovní body a referenční vstupy. Z toho plyně, že regulátor potřebuje mít informace o pracovních bodech, resp. pracovních oblastech řízeného systému. Často ale nejsme schopni určit pracovní body systému, takže FMRLC nezná přesně, které pracovní body (pracovní oblasti) se má naučit. S tím souvisí i to, že regulátor nepozná, kdy se vrátí přesně do daného pracovního bodu. Nepochybň, pokud má fuzzy regulátor dobré informace o pracovních bodech systému, tak FMRLC bude schopen se naučit a použít lepší akční zásahy. V případě, že o tom dobré informace nemá, bude se stále adaptovat, ale nebude si pamatovat, a proto se nebude učit.

Pro některé systémy mohou hodnoty  $e(k)$  a  $ce(k)$  na vstupech pouze velmi hrubě charakterizovat pracovní bod systému. V takové situaci pochopitelně není FMRLC schopen se naučit různé regulátory pro různé pracovní body. Bude vycházet pouze z limitovaných informací o pracovním bodě a bude se stále adaptovat, aby našlo co nejlepší regulátor. V tomto případě degraduje učící se mechanismus na adaptivní, který není schopen si pamatovat. To ovšem neznamená, že by se tím musela regulace nutně zhoršit. Tato situace bude dále ukázána při řízení konkrétního modelu.

Obecně se tudíž snažíme vybrat takové vstupy, aby co nejvíce určovali stav systému, ve kterých chceme, aby se regulátor naučil. To by mělo být pravidlo pro výběr vstupů do fuzzy regulátoru pro praktické aplikace. Konkurenční cíl je však udržovat počet vstupů do fuzzy regulátoru stále co nejmenší, a to z důvodu výpočetní složitosti. Tuto nevýhodu lze celkem dobře kompenzovat použitím více fuzzy regulátorů s méně vstupy ke každému z nich. Výsledný akční zásah pak můžeme v některých případech použít jako součet z jednotlivých výstupů.

#### 4.2.2 Referenční model

Dále musíme rozhodnout, jaký vybrat referenční model, který určuje požadované vlastnosti celého zpětnovazebního systému. Většinou požadujeme co nejlepší průběh regulace, ale zároveň tento průběh musí splňovat rozumné požadavky. Když budeme požadovat příliš mnoho, tak toho regulátor nebude schopen dosáhnout. Charakteristiky reálného systému určují praktická omezení, kterých chceme ještě dosáhnout. Nemusí být vždy příliš jednoduché vybrat správný referenční model, protože je občas obtížné znát, co od systému můžeme očekávat. Obecně může být referenční model lineární, nelineární, časově variantní, časově invariantní atd.

Mějme zvolený referenční model charakterizující kritéria jako je doba náběhu (tzv. *rise - time*) nebo maximální překmit (tzv. *overshoot*) a k tomu referenční signál  $r(k)$ . Požadované průběhy řízeného systému nastávají tehdy, když učící se mechanismus zmodifikuje regulátor tak, že „dotlačí“ průběh  $y_e(k)$  k velmi malé hodnotě po celou dobu regulace bez ohledu na to, který referenční signál do celého řídicího systému právě vchází. Chyba  $y_e(k)$  nám poskytuje informaci o tom, jak jsme daleko od požadovaného průběhu. Je určena podle vztahu

$$y_e(k) = y_m(k) - y(k).$$

V případě, že  $y_e(k)$  je malé, pak učící se mechanismus nebude fuzzy regulátor významně modifikovat. Na druhé straně, když  $y_e(k)$  je velké, tak jsme požadovaného průběhu nedosáhli a učící se mechanismus musí fuzzy regulátor modifikovat hodně.

### 4.2.3 Učící se mechanismus

Učící se mechanismus ladí výstupní funkce příslušnosti spolu s bází pravidel fuzzy regulátoru tak, aby se zpětnovazební smyčka chovala jako výstup z referenčního modelu  $y_m$ . Modifikace báze pravidel se provádí na základě pozorování dat z řízeného systému, referenčního modelu a fuzzy regulátoru. Učící se mechanismus obsahuje dvě hlavní částí:

1. fuzzy inverzní model
2. modifikátor báze znalostí

Fuzzy inverzní model vykonává funkci zobrazení z  $y_e(k)$  (reprezentující odchylku od požadovaného chování) na  $p(k)$ , které určuje míru modifikace pravidla ve fuzzy regulátoru, který se snaží dosáhnout  $y_e(k)$  blízké nule. Modifikátor báze znalostí vykonává funkci modifikace báze pravidel ve fuzzy regulátoru. Jednotlivé části budou teď vysvětleny detailněji.

#### Fuzzy inverzní model

Pro vytvoření fuzzy inverzního modelu potřebujeme mít alespoň hrubé znalosti o řízení systému. V našem případě potřebujeme určit zobrazení z  $y_e(k)$  a  $y_c(k)$  ( $y_c(k) = \frac{y_e(k) - y_e(k-1)}{T}$ ) na nezbytné změny  $p(k)$ . Takový fuzzy systém se nazývá *fuzzy inverzní model*, protože informace o inverzní dynamice systému je určena právě zde.

Tento model je podobný fuzzy regulátoru, fuzzy inverzní model je vidět na obr.10 a zesílení tohoto modelu jsou  $g_{y_e}$ ,  $g_{y_c}$  a  $g_p$ . Vstupy do fuzzy inverzního modelu jsou pak následující  $y_e(k)$  a  $y_c(k)$ .

Báze pravidel fuzzy inverzního modelu má následující formu:

$$\text{Když } y_e \text{ je } Y_e^j \text{ A } y_c \text{ je } Y_c^l \text{ Potom } p \text{ je } P^m$$

$Y_e^j$  a  $Y_c^l$  určuje  $j$ -tou a  $l$ -tou fuzzy množinu asociovanou se vstupem  $y_e$ , respektivě  $y_c$ ,  $P^m$  známená výstup z inverzního modelu,  $P^m$  určuje  $m$ -tou fuzzy množinu, která odpovídá výstupu  $p$ .

#### Modifikátor báze znalostí

Nechť máme danou informaci o nezbytných změnách regulátoru, které jsou reprezentovány hodnotou  $p(k)$ . Modifikátor báze znalostí změní bázi pravidel ve fuzzy regulátoru tak, že předchozí akční zásah, který způsobil neuspokojivé  $y_e$  modifikuje o  $p(k)$ . Předpokládejme, že předchozí spočítaný akční zásah byl  $u(k-1)$  a dále, že přispěl k dobré nebo špatné regulaci. V případě špatné regulace se  $y$  neblíží k  $y_m$ . Pro jednoduchost zatím uvažujme, že jsme schopni u nějakého systému v jednom kroce  $T$  ovlivnit výstup  $y$ . Připomeňme dál, že  $e(k-1)$  a  $ce(k-1)$  byla odchylka, respektive difference odchylky, které vstupovaly do fuzzy regulátoru v daném čase  $k-1$ . Modifikováním báze pravidel ve fuzzy regulátoru „nutíme“ regulátor k tomu, aby příště ve stejném stavu systému produkoval výstup  $u(k-1) + p(k)$ , který měl být již v čase  $k-1$ , aby bylo  $y_e(k)$  menší. Potom, když dostane fuzzy regulátor na vstupech podobnou odchylku  $e(k)$  a diferenční odchylku  $ce(k)$ , tak bude vstup do řízeného systému již modifikovaný akční zásah, který by měl výstup systému  $y(k)$  více přiblížit k výstupu z referenčního modelu  $y_m(k)$ .

Jako výstupní funkce regulátoru jsou symetrické trojúhelníkové funkce příslušnosti. Střed výstupní funkce příslušnosti si označíme jako  $b_m$ , který bude odpovídat dané výstupní fuzzy množině  $\tilde{U}^m$ . Modifikátor báze znalostí modifikuje regulátor tím, že posouvá středy  $b_m$  těchto funkcí příslušnosti  $\tilde{U}^m$ , která je asociována s pravidlem fuzzy regulátoru, který způsobil předchozí akční zásah  $u(k-1)$ . Zde jsou uvedeny dva kroky tohoto procesu:

1. Nalezneme všechna pravidla ve fuzzy regulátoru, která splňovala

$$\mu_i(e(k-1), c(k-1)) > 0 \quad (16)$$

a tato pravidla nazveme aktivní v čase  $k-1$ . Můžeme také charakterizovat aktivní fuzzy množiny pomocí indexů u jednotlivých funkcí příslušnosti. Používáme všechny možné kombinace pravidel a ke každé kombinaci bude příslušet jedna výstupní funkce příslušnosti.

2. Stanovíme  $b_m(k)$  určující střed  $m$ -té výstupní funkce příslušnosti v čase  $k$ . Pro všechna aktivní pravidla použijeme

$$b_m(k) = b_m(k-1) + p(k) \quad (17)$$

pro modifikaci středů výstupních funkcí příslušnosti. Pravidla, která nebyla aktivní, nebudou během modifikace pozměněna.

Je potřeba zdůraznit, že v našem případě modifikace zajistí to, že v dalších časových okamžicích bude pro stejnou odchylku  $e(k-1)$  a diferenci odchylky  $ce(k-1)$  již akční zásah  $u(k-1)+p(k)$ . To ovšem neznamená, že nutně musí dojít ke zlepšení regulace, protože se systém může nacházet v jiném stavu, který tyto dvě hodnoty  $e(k-1)$  a  $ce(k-1)$  nedokáží rozlišit. To ale souvisí s omezeným použitím již samotného PD regulátoru.

Pro náš případ, kdy fuzzy regulátor má vstupní trojúhelníkové funkce příslušnosti rovnoměrně rozmištěny, jak je zobrazen na obr.12, může nastat situace taková, že mohou být maximálně dvě funkce příslušnosti aktivní na každém vstupu. Čili pro dva vstupy mohou být maximálně čtyři funkce příslušnosti aktivní v čase  $k-1$ , tudíž platí vztah  $\mu_i(e(k-1), c(k-1)) > 0$ . Z toho vyplývá, že nanejvýš čtyři pravidla, resp. výstupní funkce příslušnosti, mohou být modifikována.

Příklad na modifikaci báze pravidel je ukázán na konkrétním příkladu. Pro názornost předpokládejme, že všechna zesílení do fuzzy regulátoru, tj. $g_e$ , a  $g_c$ , stejně tak zesílení do fuzzy inverzního modelu  $g_{ye}$ ,  $g_{yc}$  jsou rovné hodnotě 1. Předpokládejme, že fuzzy inverzní model produkuje výstup  $p(k) = 0.5$ , což nám říká, že hodnota akčního zásahu v čase  $k-1$  měla být  $u(k-1)+0.5$ , aby se  $ye$  přiblížilo nule. Dále předpokládejme, že  $e(k-1) = 0.75$  a  $ce(k-1) = -0.2$  a funkce příslušnosti na vstupu byly použity ty, které jsou zobrazeny na obr. 12.

V tomto případě jsou aktivní tato pravidla:

**Když**  $e(k-1)$  je  $E^3$  **A**  $ce(k-1)$  je  $C^{-l}$  **Potom**  $u(k-1)$  je  $U^1$

**Když**  $e(k-1)$  je  $E^4$  **A**  $ce(k-1)$  je  $C^{-l}$  **Potom**  $u(k-1)$  je  $U^2$

Z obr. 12 zjistíme hodnoty  $\mu_1 = 0.25$  a  $\mu_2 = 0.75$ . Pouze tato dvě pravidla modifikují posunutí fuzzy množin  $U^1$  a  $U^2$  na výstupu. Předpokládejme, že v čase  $k - T$  jsme měli  $b_1(k - 1) = 1$  a  $b_2(k - 1) = 3$ . Modifikaci těchto fuzzy množin provedeme tak, že jednoduše posuneme jejich středy podle rovnice (17):

$$b_1(k) = b_1(k - 1) + p(k) = 1 + 0.5 = 1.5$$

$$b_2(k) = b_2(k - 1) + p(k) = 3 + 0.5 = 3.5$$

### Učení, pamatování si a výběr vstupů do fuzzy inverzního modelu

Výběr vstupů do fuzzy regulátoru má tudíž základní dopad na učení a pamatování si. Totéž platí i pro vstupy do fuzzy inverzního modelu. Měli bychom vybírat takové vstupy, aby se regulátor adaptoval v různých podmínkách různě. V některých pracovních podmínkách budeme např. požadovat, aby adaptace byla pomalejší než v jiné. V jiných pracovních podmínkách budeme např. požadovat posunutí středů výstupních funkcí v opačném směru. V případě více fuzzy regulátorů budeme také požadovat více inverzních fuzzy modelů, které nám pomáhají snížit výpočetní složitost tím, že budeme používat méně vstupů na jeden inverzní fuzzy model.

Výběr vstupů do fuzzy inverzního modelu je ukázán na obr.10, což nám v tomto případě říká, že adaptace probíhá různě pro různou odchylku a různou diferenci odchylky mezi výstupem  $y$  a žádanou hodnotou  $y_m$ . Inverzní model může být navrhnut tak, že např. pro malu odchylku  $y_e$  bude modifikace fuzzy regulátoru malá a když chyba je malá, ale differenční odchylka rychle roste, musí být potom modifikace větší. Pravidla podobného typu jsou nahrána do fuzzy inverzního modelu.

### Alternativní modifikace báze znalostí

Pro připomenutí předpokládáme, že akční zásah  $u(k)$  ovlivnil výstup řízeného systému v jednom časovém okamžiku, takže platí, že výstup  $y(k + T)$  je ovlivněn akčním zásahem  $u(k)$ . Je zřejmé, že toto platí spíše výjimečně a rozhodně ne obecně. Pokud chceme zobecnit tento předpoklad, definujme si nejdříve počet časových kroků  $d$ , který znamená, že akční zásah  $u(k)$  začne mít vliv na výstup v čase  $k + d$ . Jinými slovy, výstup  $y(k + d)$  je ovlivněn vstupem akčního zásahu  $u(k)$ . Přístup je podobný vztahu (16), ale v tomto případě musíme započít ještě toto zpoždění  $d$ . Po úpravě tedy dostaneme:

$$\mu_i(e(k - d), c(k - d)) > 0, \quad (18)$$

která označuje aktivní pravidla v čase  $k - d$ . Pro modifikaci aktivních pravidel pak dostaneme následující vztah:

$$b_m(k) = b_m(k - d) + p(k) \quad (19)$$

Pokud zvolíme  $d = 1$ , dostaneme přímo vztah (16) a s tím související vztah (17). To zajistí, že modifikujeme pravidla, která aktuálně přispěla k současnemu výstupu  $y(k)$ , jehož kvalita je charakterizována  $y_e(k)$ . Zpoždění  $d$  se určuje nejsnáze u systémů s čistě dopravním zpožděním  $T_d$ , které je konstantní. V takovém případě pro určení  $d$  pustíme do soustavy krátký pulz a budeme sledovat, jak dlouho bude systému trvat, než začne na tento pulz reagovat. Pro systémy s dopravním zpožděním taková volba funguje celkem spolehlivě.

Obecně je však obtížné toto zpoždění  $d$  určit. Jedná se o tentýž problém jako v případě samoorganizujícího se fuzzy regulátoru v sekci 4.1. Občas je potřeba vyzkoušet několik takových  $d$  přímo na simulaci a na základě toho pak vhodné  $d$  vybrat. Hodnota  $d$  nám říká, jak hodně „zpátky do minulosti“ chceme jít, aby chom penalizovali. Žádný analytický vztah pro určení  $d$  neexistuje.

Existují i další alternativy modifikace báze znalostí, které souvisí s volbou zpoždění  $d$ . Rovnici (18) můžeme zapsat také jako

$$\mu_i(e(k-d), c(k-d)) > \alpha, \quad (20)$$

kde  $0 \leq \alpha < 1$ . V tomto případě nebudou modifikována pravidla, u kterých fuzzifikovaná hodnota je pod zadanou mezí  $\alpha$ . To nám pomůže v případě, kdy nechceme modifikovat pravidla, u kterých vychází v předpokladech fuzzifikované hodnoty pod danou mez  $\alpha$ . Jsou to pravidla, u kterých si nejsme moc jisti, zda by měla být použita. Tento přístup je zvlášť výhodný při použití Gaussovských funkcí příslušnosti na vstupu, protože bude zajištěno, že nebude docházet k modifikaci všech středů funkcí na výstupu v každém časovém kroce, a proto lokální učící se charakteristika bude zachována.

Vztah (19) můžeme také poupravit na následující

$$b_m(k) = b_m(k-d) + \mu_m((e(k-d), c(k-d)) > \alpha) * p(k), \quad (21)$$

která nám vyjadřuje, že posun středu výstupní funkce příslušnosti bude záviset na fuzzifikované hodnotě v předpokladu. To je smysluplné, protože budeme měnit výstupní funkce příslušnosti pravidel tím více, čím je více „zřetelné“, která funkce na vstupu byla v daný okamžik aktivní. Tento přístup je více efektivní než nám dává vztah (17). Obecně to ale opět neplatí, ale je to výhodné v případě, kdy se objevují nežádoucí oscilace v systému, které jsou způsobeny nadměrně častým použitím modifikace, tedy posuny středů výstupních funkcí příslušnosti.

Další modifikaci, kterou můžeme do algoritmu přidat, je zajištění středů výstupních funkcí v žádaném rozmezí, což je velmi důležité především pro praktické aplikace. V některých případech můžeme např. požadovat, aby středy na výstupu byly vždy kladné, takže regulátor nebude nikdy poskytovat záporný akční zásah. Jindy je potřeba udržovat absolutní hodnoty těchto středů stále menší než nějakou předem určenou hodnotu, takže výstup z regulátoru nebude větší, resp. menší, než zadaná mez, což je velice důležité pro praktické aplikace, kdy jsme vždy omezeni maximálním možným akčním zásahem.

Předpokládejme, že známe předem rozsah dovolených středů výstupních funkcí příslušnosti  $[b_{min}, b_{max}]$ . Toho dosáhneme tím, že přidáme tato dvě pravidla:

$$\begin{aligned} \textbf{Když } b_m(k) < b_{min} \textbf{ Potom } b_m(k) &= b_{min} \\ \textbf{Když } b_m(k) > b_{max} \textbf{ Potom } b_m(k) &= b_{max} \end{aligned}$$

Jinými slovy, pokud středy překročí stanovenou mez, jsou nastaveny na tato omezení. Někdy však nevíme, jak výkonný akční člen vlastně potřebujeme a až na základě výsledků simulací vhodný zakoupíme. Pak je dobré tuto funkci odstranit a nechat celou simulaci adaptace „projekt“ a následně se pak podívat, jaký akční zásah vychází a podle toho vhodný aktuátor zakoupit.

Existují i další možnosti, jak bázi znalostí modifikovat. Záleží hodně na konkrétní aplikaci. Můžeme např. část pravidel „zablokovat“, takže nebudou během adaptace

vůbec modifikována. To může být prospěšné, pokud už tato část regulátoru bude správně naučena, jak systém řídit. Pokud chceme být obezřetní, můžeme čekat např.  $d$  nebo více časových kroků před další adaptací. Tímto si tak ověříme, že daná modifikace skutečně „pomohla“ a vyhneme se tak případným nežádoucím oscilacím, když se trochu změní pracovní bod.

### Pravidla pro návrh inverzního fuzzy modelu

Zde je uvedeno pár pravidel, které slouží pro návrh fuzzy inverzního modelu a jeho zesílení  $y_e$ ,  $y_c$ ,  $g_p$ . Neexistuje zatím žádná obecná metoda, velmi záleží na konkrétní aplikaci. Existuje však přesto pár pravidel, které v řadě případů pomáhají a vyplatí se je proto znát.

Je potřeba si uvědomit, že pro množství aplikací je najít fuzzy inverzní model zhruba stejně obtížné jako návrh samotného fuzzy regulátoru. Fuzzy inverzní model má často symetrické vlastnosti a stejnou strukturu jako regulátor, který ladíme.

Dále je nezbytné definovat tento inverzní model tak, že v případě, kdy výstup systému těsně sleduje výstup z modelu  $y_m$ , musí být schopen vypnout adaptaci. Tímto způsobem se dostávají vstupy do fuzzy inverzního modelu těsně k nule. Zahrnutí této charakteristiky do fuzzy inverzního modelu může občas zajistit stabilitu celého zpětnovazebního systému. Jiný přístup je implementace této charakteristiky v tom, že zmodifikujeme výstup fuzzy inverzního modelu použitím následujícího pravidla:

$$\text{Když } |p(k)| < \epsilon_p \text{ Potom } p(k) = 0$$

kde  $\epsilon_p > 0$  je malé číslo, které je a priori určeno. V případě, že vstupy do inverzního fuzzy modelu jsou blízko nuly, bude tak na výstupu fuzzy inverzního modelu nula, čili nebude docházet k adaptaci, resp učení. Je to ochrana např. před šumem, který se přičítá k výstupu  $y$  a který by způsoboval, že by se adaptace snažila modifikovat regulátor tak, aby výstup dosáhl tohoto průběhu se šumem. V případě, že chyba mezi výstupem  $y$  a výstupem z modelu  $y_m$  roste, tak se učí se (resp. adaptivní) mechanismus sám zapne a bude se opět snažit vstupy do fuzzy inverzního modelu „dotlačit“ do nuly.

### Návrh zesílení fuzzy inverzního modelu

1. Vybereme zesílení  $g_{y_e}$  tak, že  $y_e(k)$  nebude v saturaci s krajní vstupní funkcí příslušnosti. Toto je velmi heuristický výběr, protože nemůžeme znát předem, jak  $y_e(k)$  bude velké, ale v některých aplikacích toto může být dobře určeno.
2. Vybereme zesílení  $g_p$  stejné jako zesílené  $g_u$  ve fuzzy regulátoru a nastavíme  $g_{y_c} = 0$ .
3. Jako referenční signál použijeme jednotkový skok  $r(k)$  o velikosti, která bude typická během celé operace.
4. Pozorujeme odezvy soustavy referenčního modelu.
  - Když se vyskytují neakceptovatelné oscilace ve výstupní odezvě systému okolo odezvy z referenčního modelu, potom zvýšíme  $g_{y_c}$  (potřebujeme dostatečný derivační zásah v učícím se mechanismu k redukování oscilací). Vrátíme se k bodu 3.

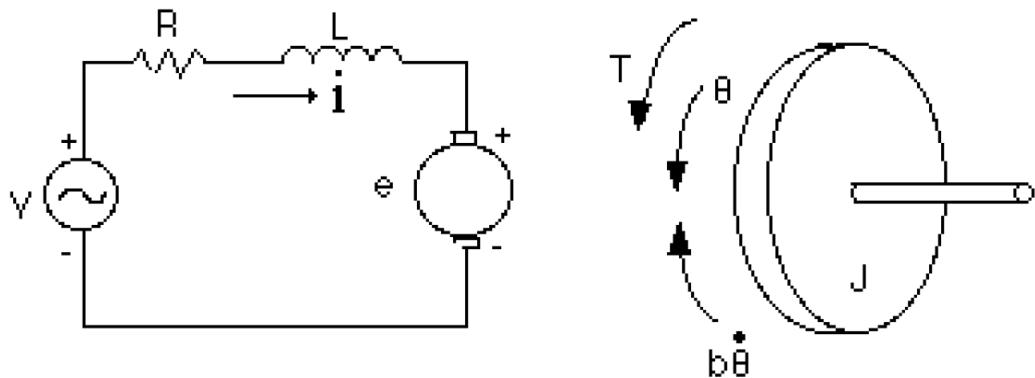
- Když výstup ze systému nemůže být schopný sledovat výstupní referenční signál, tak snížíme  $g_{yc}$ . Vrátíme se k bodu 3.
- Když je odezva systému akceptovatelná vzhledem k signálu z referenčního modelu, tak je návrh kompletní.

## 5 FMRLC na konkrétních modelech

Metoda FMRLC je postupně vyzkoušena na třech vybraných modelech. Ve všech třech případech se jedná o systémy s jedním vstupem a jedním výstupem, přičemž první model je lineární, zbylé již nelineární. Všechny simulace byly provedeny v programovém prostředí *MATLAB - SIMULINK*.<sup>1</sup>.

### 5.1 Řízení servosystému

Stejnosměrný servomotor je běžně používaný aktuátor v řídicích systémech (obr. 13). Servosystém umožňuje jak rotační, tak translační pohyb.



Obrázek 13: Schéma DC servosystému

Jeho model budeme považovat za lineární. Vztah mezi úhlem natočení  $\Theta$  a vstupním napětím  $V$  daného servosystému vyjadřuje následující přenos:

$$\frac{\Theta(s)}{V(s)} = \frac{K}{LJs^3 + (RJ + bL)s^2 + (bR + K^2)s} \quad (22)$$

#### Parametry systému

- moment setrvačnosti rotoru:  $J = 3 * 10^{-6}$  kg\*m<sup>2</sup>/s<sup>2</sup>,
- koeficient tlumení mechanického systému:  $b = 4.5 * 10^{-4}$  N\*s/m,
- koeficient síly:  $K = 0.03$  N\*m/A,
- rezistence cívky:  $R = 4 \Omega$ ,
- elektrická indukčnost:  $L = 2.0 * 10^{-6}$  H.

Po dosazení parametrů a úpravě dostaneme následující přenos:

$$\frac{\Theta(s)}{V(s)} = \frac{12.5}{(1 + 5 * 10^{-3}s)(1 + 5 * 10^{-7}s)s} \quad (23)$$

---

<sup>1</sup>MATLAB, SIMULINK jsou ochranné známky společnosti Math Works, Inc.

### 5.1.1 Výsledky simulací servosystému

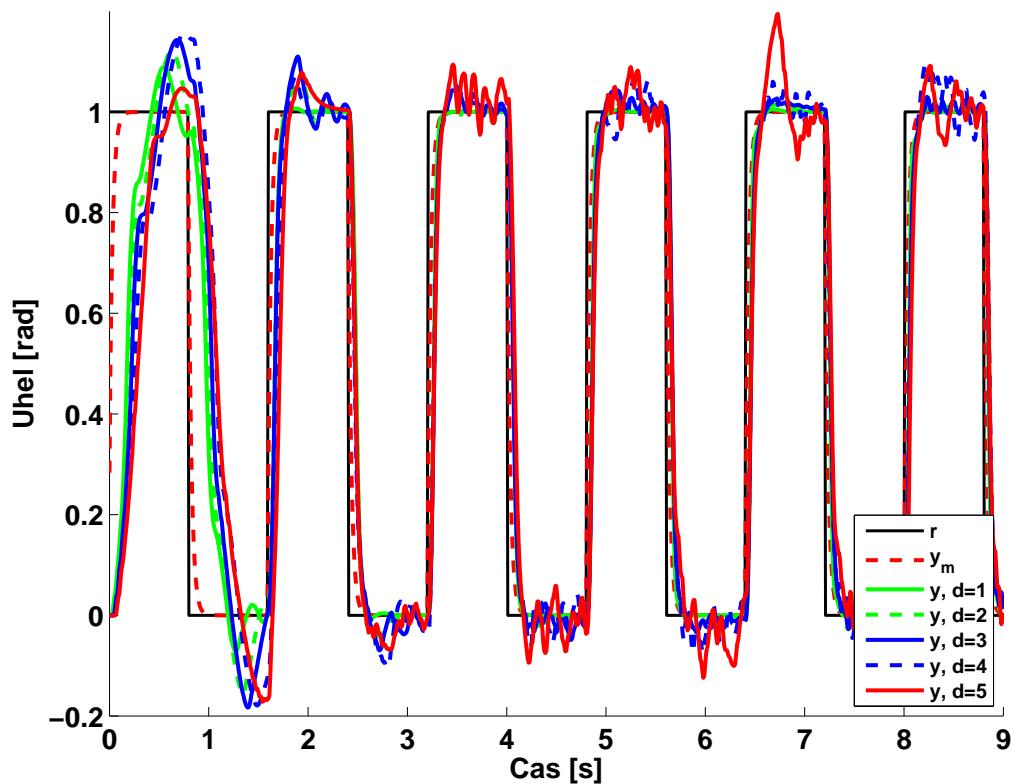
Cílem řízení je sledování požadovaného úhlu natočení. Na každém ze dvou vstupů bylo zvoleno 11 trojúhelníkových funkcí příslušností, které jsou rovnoměrně rozmištěny. Jsou brány v úvalu všechny možné kombinace, tedy na výstupu může být až  $11 * 11 = 121$  různých trojúhelníkových funkcí. Regulátor má v počátečním stavu všechny středy výstupních funkcí rovné nule, nemá tak žádné informace o tom, jak daný systém řídit. V učícím se mechanismu v inverzním modelu jsou pouze přibližná pravidla, podle kterých se v jednotlivých stavech systému regulátor modifikuje za účelem dosažení co nejlepší regulace. Akční zásah je omezen na  $\pm 1V$ . Vzorkovací perioda je nastavena na  $T = 0.01$  s. FMRLC bylo vytvořeno v souladu se schématem na obr.10.

Na obr.14 jsou zobrazeny průběhy sledování požadovaného úhlu při zpožděních  $d = 1, 2, 3, 4$  a  $5$ , na obr.15 jsou zobrazeny průběhy sledování požadovaného úhlu při zpožděních  $d = 6, 7, 8, 9$  a  $10$ . Z nich je vidět, že nejvhodnější volba zpoždění je  $d = 1$ , protože se vzrůstajícím  $d$  se regulace výrazně zhoršuje. Je to z důvodu toho, že se jedná o velice rychlý systém, který na akční zásahy reaguje téměř okamžitě a právě tuto vlastnost nejlépe charakterizuje nejmenší z vyzkoušených zpoždění  $d = 1$ . Byly vyzkoušeny dva různé referenční signály (na obr.16 a 18 obrázky horní vlevo). Na oba průběhy se FMRLC naučil dobře regulovat. Na průběh referenčního signálu ve tvaru pravoúhlých pulzů trvala adaptace zhruba 5 s, protože největší penalizace byly právě do prvních 5 s (obr.17 dolní vlevo). Na průběh referenčního signálu ve tvaru sinu trvala adaptace ještě méně, pouze 1 s. Penalizace od 1 s dále klesala v poměru k penalizaci na počátku na ještě méně významnou velikost (obr.19 dolní vlevo) než v případě referenčního signálu ve tvaru pravoúhlých pulzů.

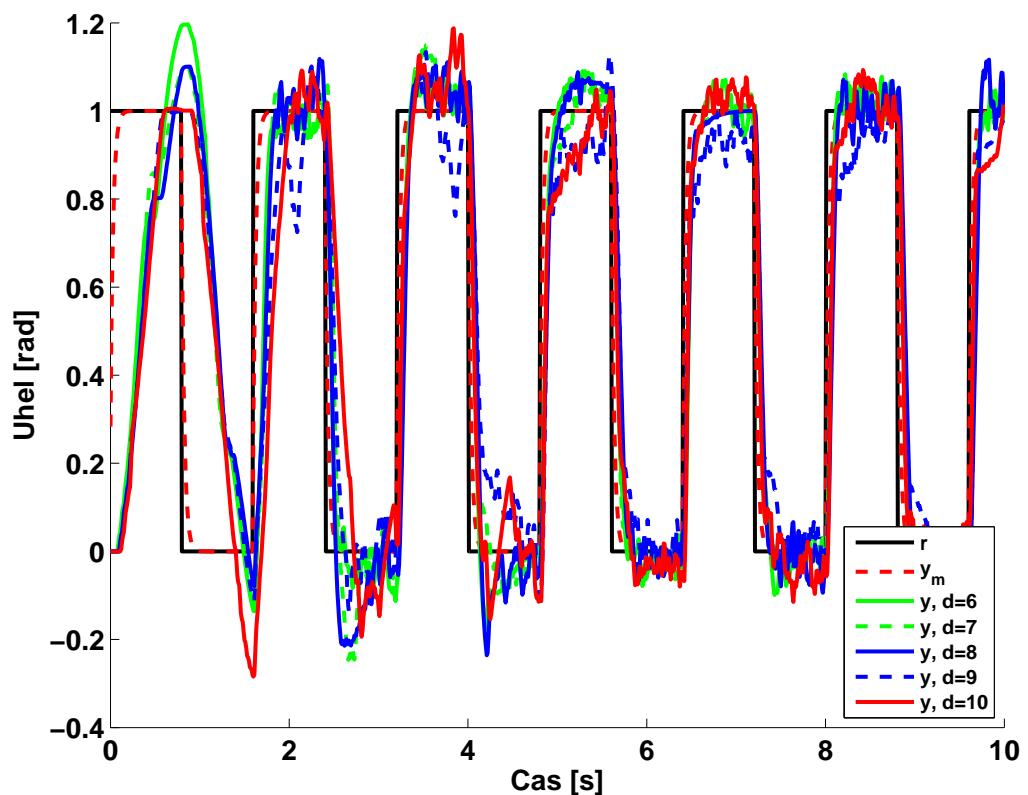
Pro oba průběhy vyšly výsledné báze pravidel trochu odlišně (obr. 16 vpravo dole a obr.18 vpravo dole), což je způsobeno tím, že např. při referenčním signálu pravoúhlého průběhu se systém nedostal do některých stavů, do kterých se dostal při referenčním signálu tvaru sinu. Z tohoto důvodu kdybychom chtěli jednu z těchto bází použít do pevného fuzzy regulátoru, takový regulátor nebude fungovat pro jiné referenční signály. Řešením je použít více referenčních signálů, na které by se stále stejná báze pravidel postupně adaptovala. Klíčové je, aby se systém dostal do všech možných stavů a byla tak navštívěna (co možná nejvícekrát) všechna polička v bázi pravidel představující středy výstupních funkcí příslušnosti.

Na obr.20 je ukázána situace, kdy se náhle změní parametry řízeného systému. Z obrázku je vidět, že i s náhlou změnou parametrů systému se FMRLC algoritmus dokáže vyrovnat.

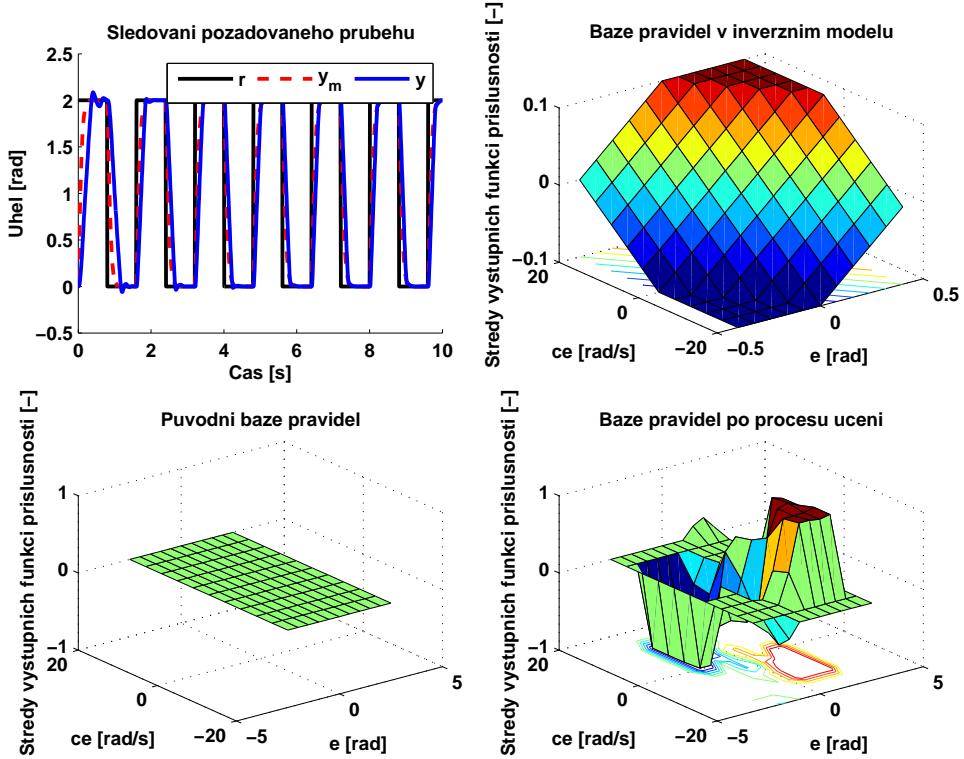
Pro lineární systémy tedy FMRLC algoritmus pracuje velice dobře. Výhoda této metody je však v tom, že je použitelná především pro plně nelineární systémy, proto další modely již budou pouze nelineární.



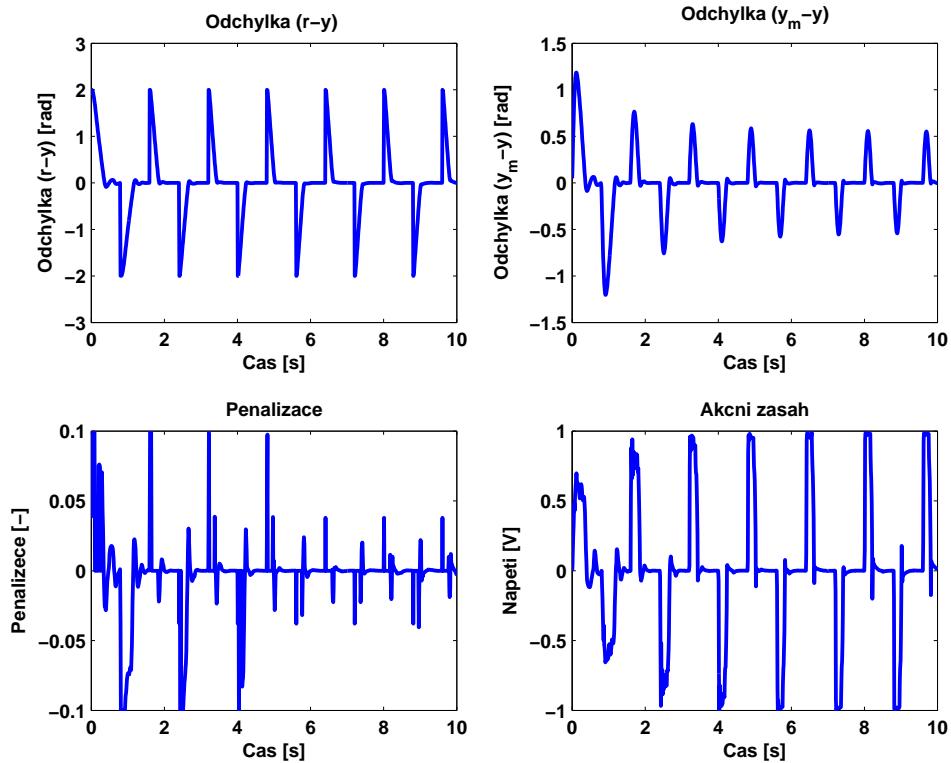
Obrázek 14: Vliv „malých“ zpoždění  $d$  na regulaci servosystému



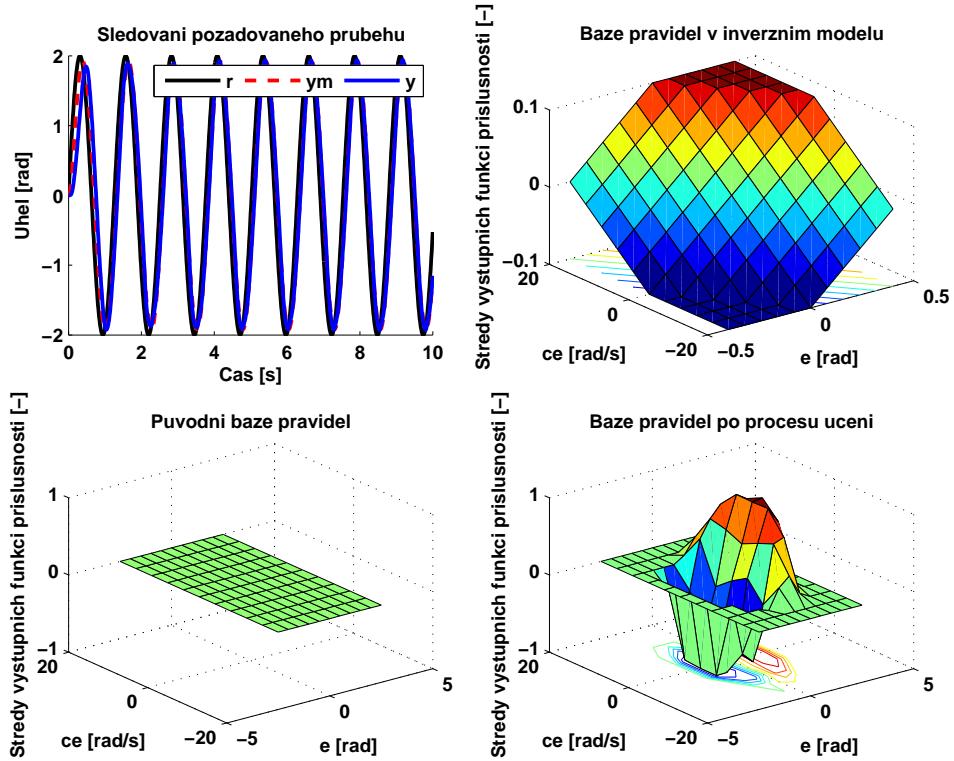
Obrázek 15: Vliv „velkých“ zpoždění  $d$  na regulaci servosystému



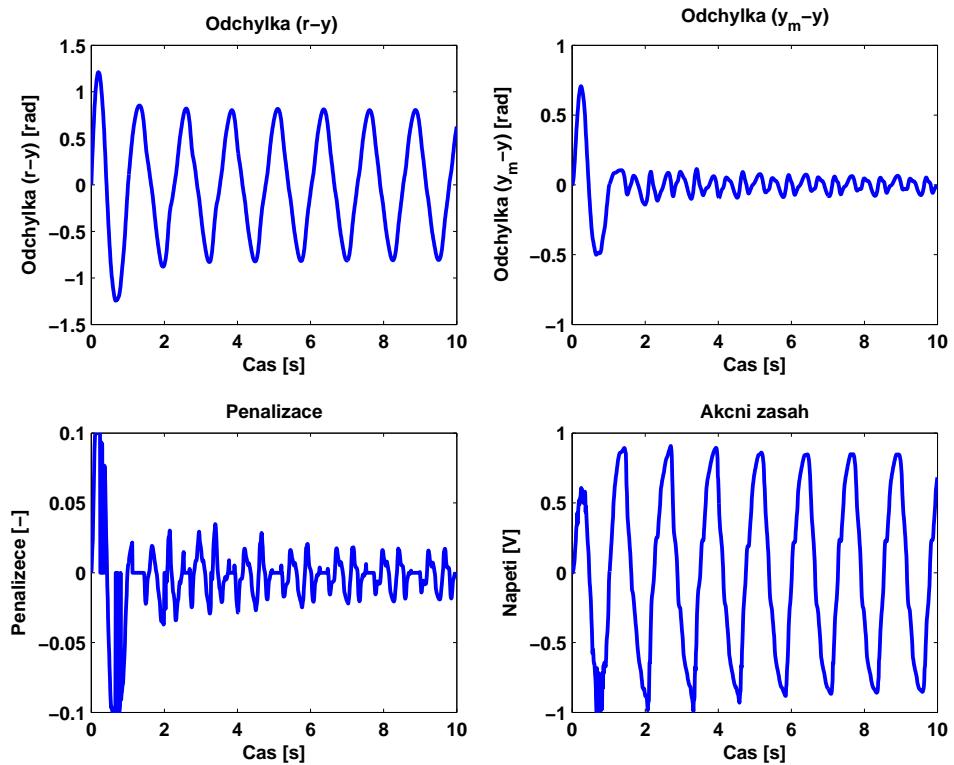
Obrázek 16: Modifikace báze pravidel regulátoru po procesu učení při sledování požadovaného průběhu, zpoždění  $d = 1$



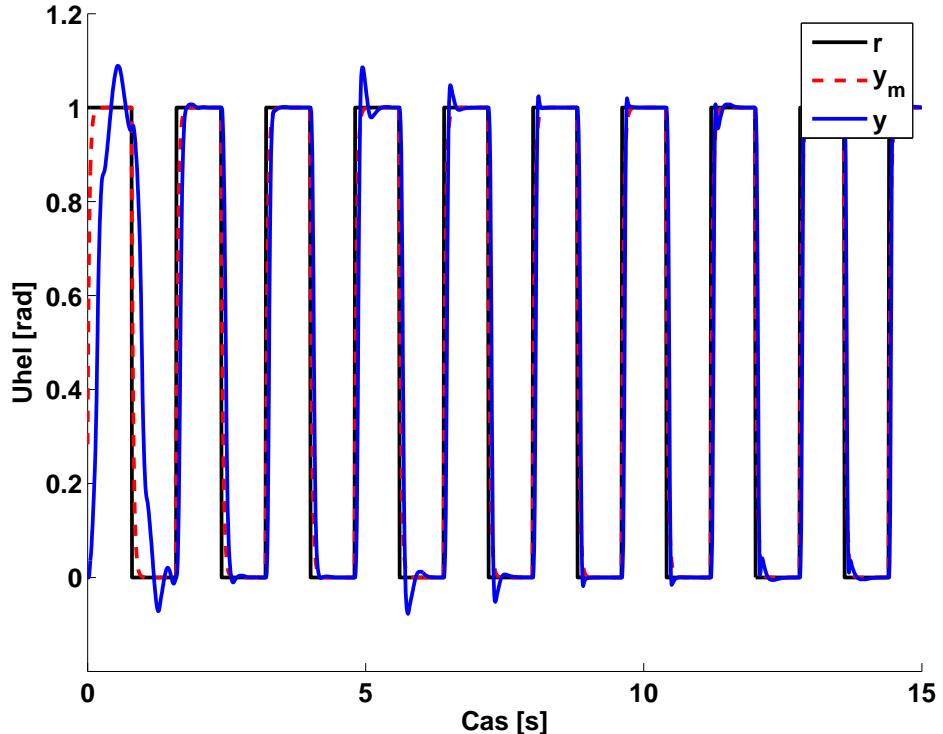
Obrázek 17: Jednotlivé průběhy veličin regulace vztažené k předchozímu obr.16



Obrázek 18: Modifikace báze pravidel regulátoru po procesu učení při sledování požadovaného průběhu, zpoždění  $d = 1$



Obrázek 19: Jednotlivé průběhy veličin regulace vztažené k předchozímu obr.18



Obrázek 20: Sledování požadovaného průběhu při náhlé změně parametrů řízeného systému v čase  $t = 4.8s$  (změna z původních parametrů  $b = 4 * 10^{-4}$ ,  $J = 3 * 10^{-6}$  na  $b = 4 * 10^{-5}$ ,  $J = 4 * 10^{-6}$ )

## 5.2 Řízení magnetické levitace

Cílem magnetické levitace je udržovat kuličku v požadované poloze mezi zemí a cívkou. Pro jednoduchost uvažujeme pouze vertikální směr. Celá situace je zobrazena na obr. 21. Vstupem systému je napětí  $v(t)$  do cívky a výstupem je poloha kuličky  $y(t)$ .

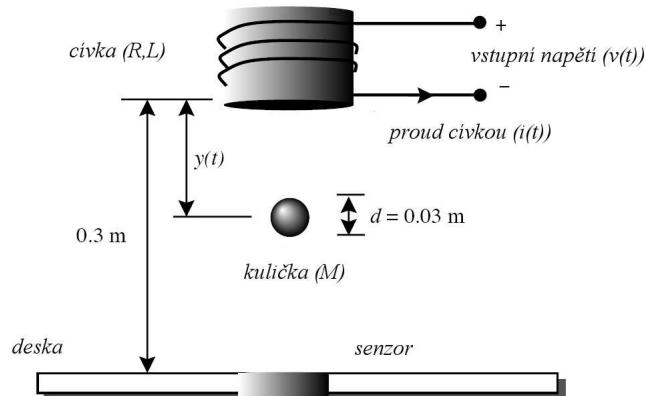
Popis systému ve stavovém popisu následující:

$$\begin{aligned} \frac{dx_1}{dt} &= x_2(t) \\ \frac{dx_2}{dt} &= g - \frac{x_3^2(t)}{Mx_1(t)} \\ \frac{dx_3}{dt} &= -\frac{R}{L}x_3(t) + \frac{1}{L}v(t), \end{aligned} \quad (24)$$

kde  $[x_1(t), x_2(t), x_3(t)]^T = [y(t), \frac{dy(t)}{dt}, i(t)]^T$ .

Význam jednotlivých proměnných je následující:

$y(t)$  [m] je pozice magnetické kuličky,  $M = 0.1$  kg je hmotnost kuličky,  $g = 9.81$  m/s<sup>2</sup> je gravitační zrychlení,  $R = 30 \Omega$  je rezistence cívky,  $L = 0.4$  H je indukčnost cívky,  $v(t)$  je vstupní napětí,  $i(t)$  je aktuální proud. Pozice kuličky je detekována např. fotorezistivním senzorem a předpokládáme, že je plně funkční v celém rozsahu polohy kuličky mezi zemí a cívkou.



Obrázek 21: Model magnetické levitace

### 5.2.1 Výsledky simulací řízení magnetické levitace

Podobně jako v příkladu servosystému byly i zde použity na každém ze dvou vstupů regulátoru 11 trojúhelníkových funkcí příslušnosti, které jsou rovnoměrně rozmištěny. Regulátor má v počátečním stavu všechny středy výstupních trojúhelníkových funkcí rovny nule, proto opět nemá žádné informace o tom, jak systém řídit. Má pouze opět přibližná pravidla obsažená v inverzním modelu, podle kterých v jednotlivých stavech systému modifikuje regulátor. Vzorkovací perioda je nastavena shodně na  $T = 0.01$  s.

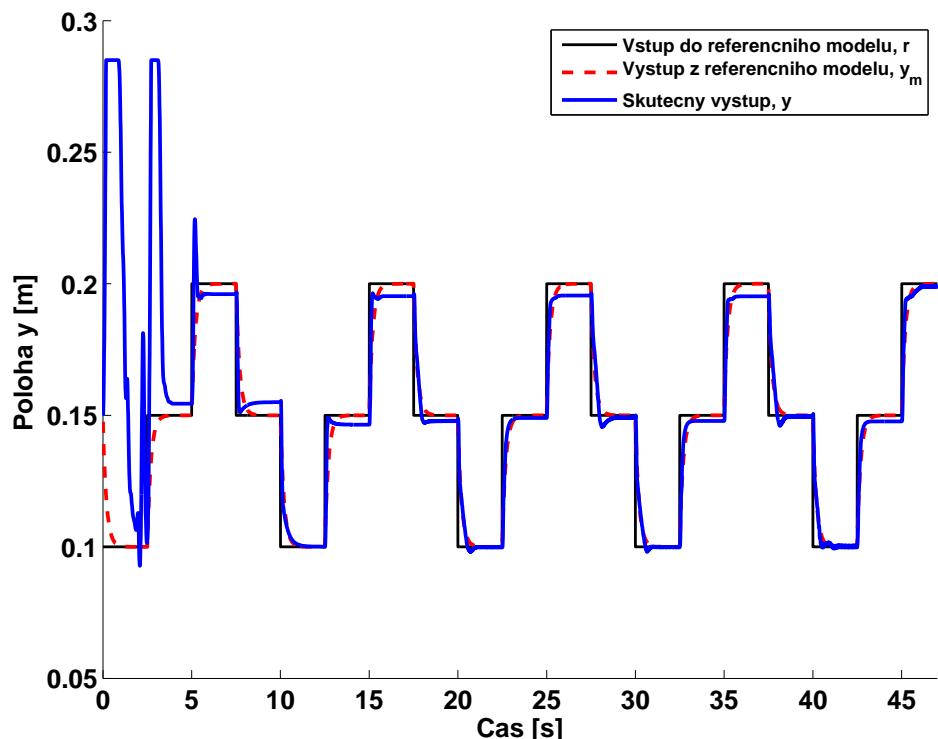
FMRLC metoda použitá na systému magnetické levitace představuje řízení ne-lineárního systému. Stejně jako u servosystému, tak i u magnetické levitace nemá regulátor na začátku žádné informace o tom, jak systém řídit. V obou těchto případech byla těmto systémům dodána stejná hrubá pravidla řízení obsažená ve fuzzy inverzním modelu. Zesílení byla zvolena pro každý model individuálně, ale pravidla zůstala stejná.

Nejdříve byla provedena regulace podle schématu na obr.10. Výsledek je zobrazen na obr.22, ze kterého je vidět, že se jedná o velmi špatnou regulaci. I tak se ale jedná o nejlepší regulaci, které bylo tímto schématem dosaženo. Jako důvod se zdá být to, že klasická zpětná vazba regulačního obvodu z obr.10 se snaží minimalizovat odchylku  $e = r - y$ , zatímco vnější smyčka s učícím se mechanismem modifikuje fuzzy regulátor tak, aby minimalizoval odchylku  $y_e = y_m - y$ . Jedná se tak vlastně o dva rozdílné cíle, které nemohou být zároveň splněny. Z tohoto důvodu bylo autorem této práce modifikováno schéma na již zmíněné schéma na obr.11, které bylo použito pro všechny následující simulace.

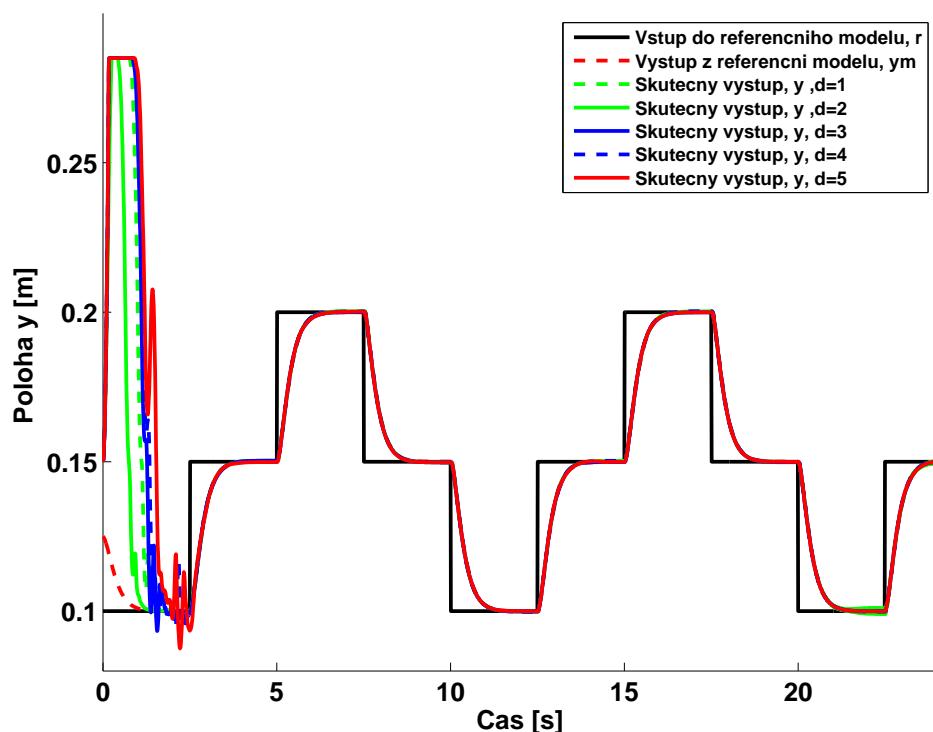
Na obr.23 a 24 je ukázán vliv zpoždění  $d$  na regulaci podobně jako u předchozího servosystému. V tomto případě vychází nejlépe zpoždění  $d = 4$ , které bylo proto použito pro všechny následující simulace.

Na obr.25 a obr.27 jsou zobrazeny dva požadované průběhy včetně svých výsledných bázi pravidel, které si jsou podobné. Na obr.26 je zobrazen mimo jiné průběh odchylky  $y_m - y$ , ze kterého je vidět, že po zhruba 1 s je maximální ochylka od požadovaného průběhu asi 3 mm. Sledování požadovaného průběhu je tak velmi přesné. Penalizace sice probíhá po celou dobu regulace, ale je potřeba si uvědomit, že se pak už jedná pouze o velmi malé hodnoty, které opět už nemají žádný výrazný vliv na výslednou bázi pravidel.

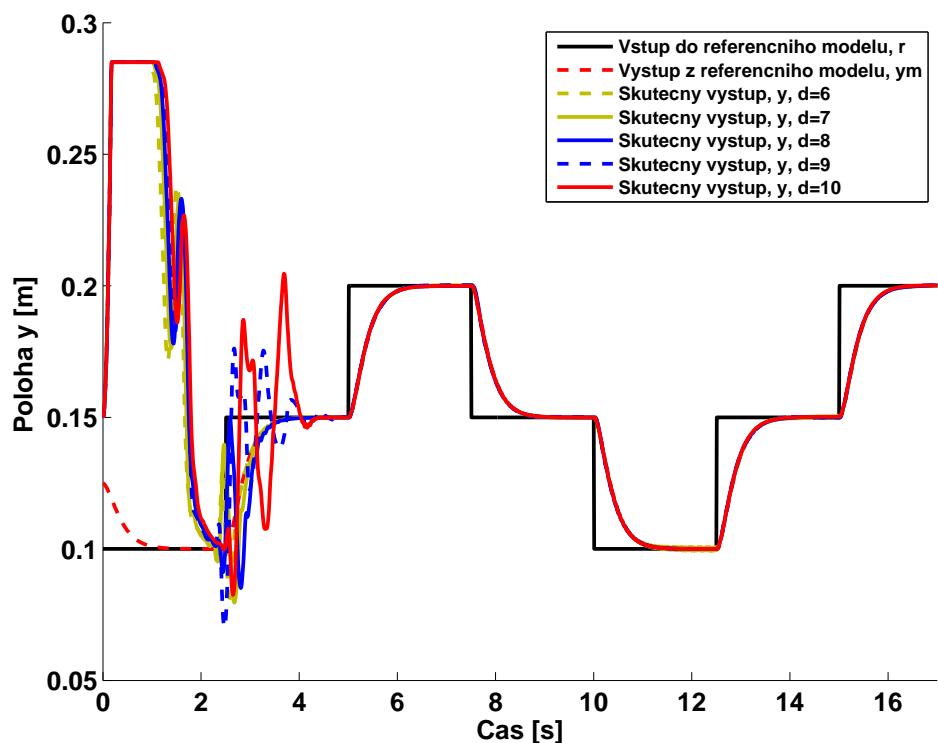
Z uvedených výsledků vyplývá, že regulace je velmi úspěšná a pro různé referenční signály jsme získali podobné výsledné modifikované báze pravidel.



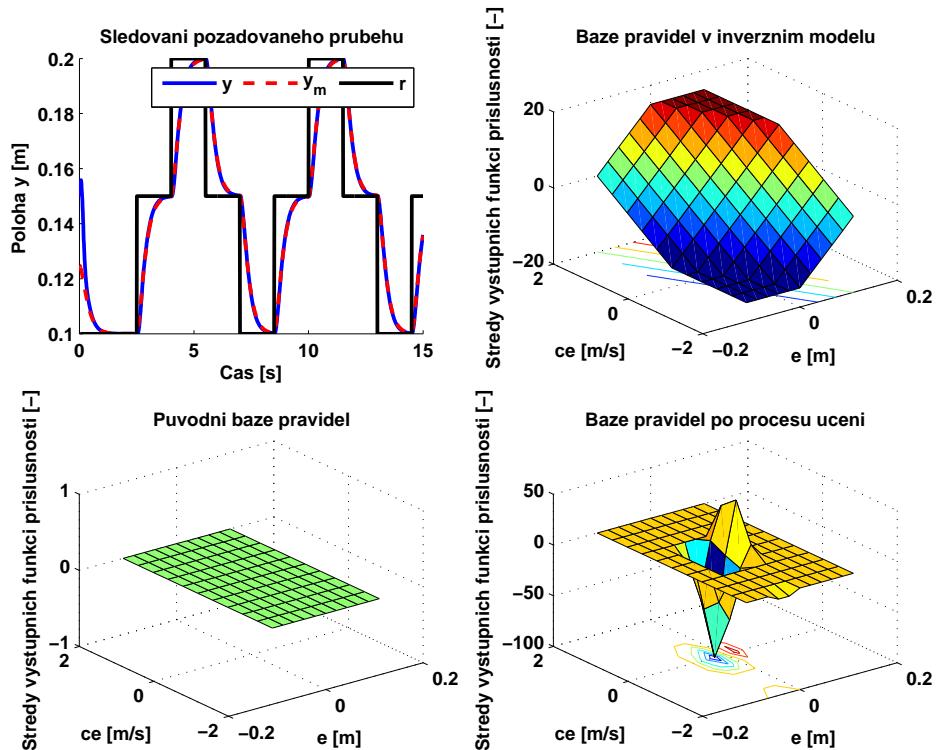
Obrázek 22: Nejlépe dosažený průběh regulace  
(provedeno podle schématu z obr.10)



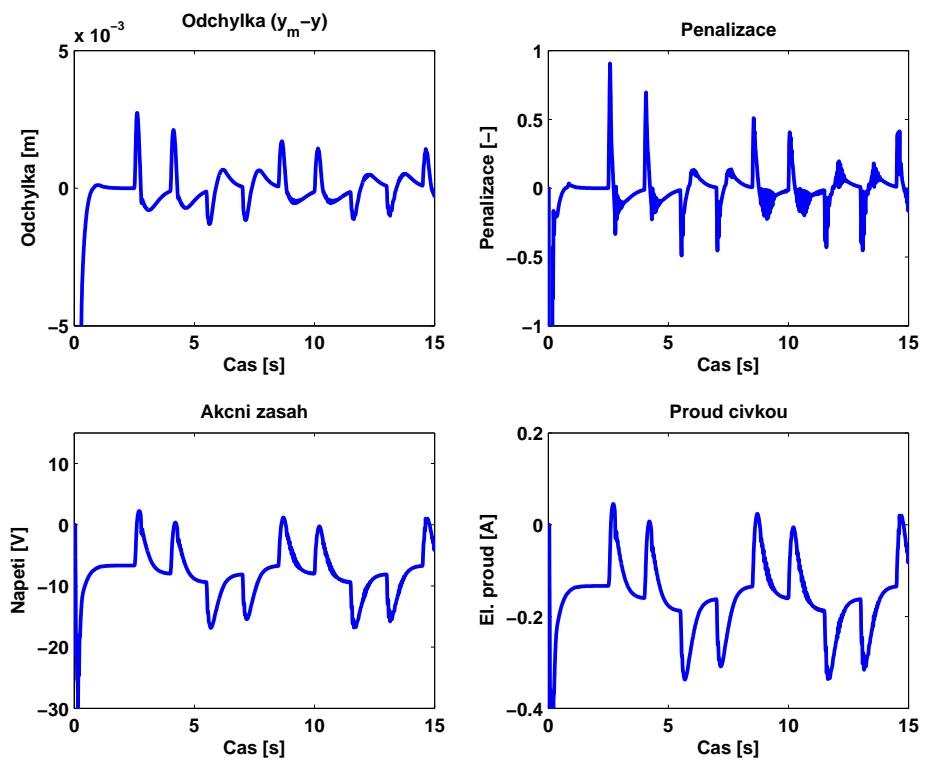
Obrázek 23: Vliv různých zpoždění na adaptaci pro zobrazený referenční průběh  
(provedeno podle schématu z obr.11)



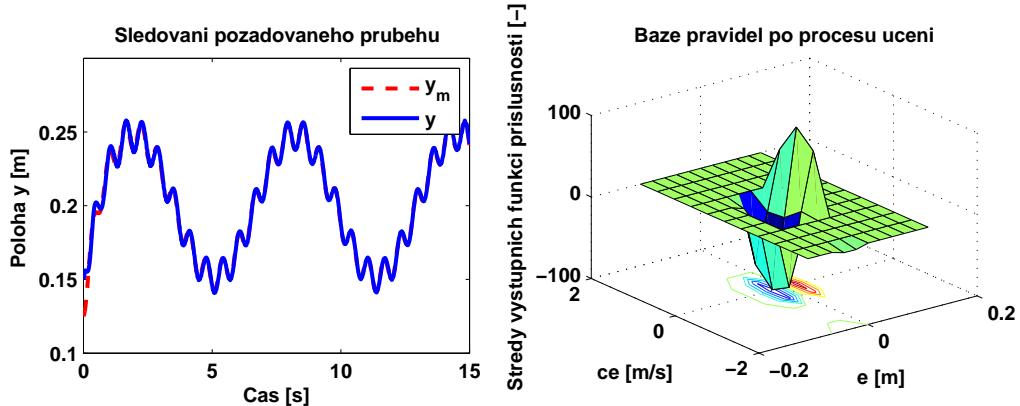
Obrázek 24: Vliv různých zpoždění na adaptaci pro zobrazený referenční průběh (provedeno podle schématu z obr.11)



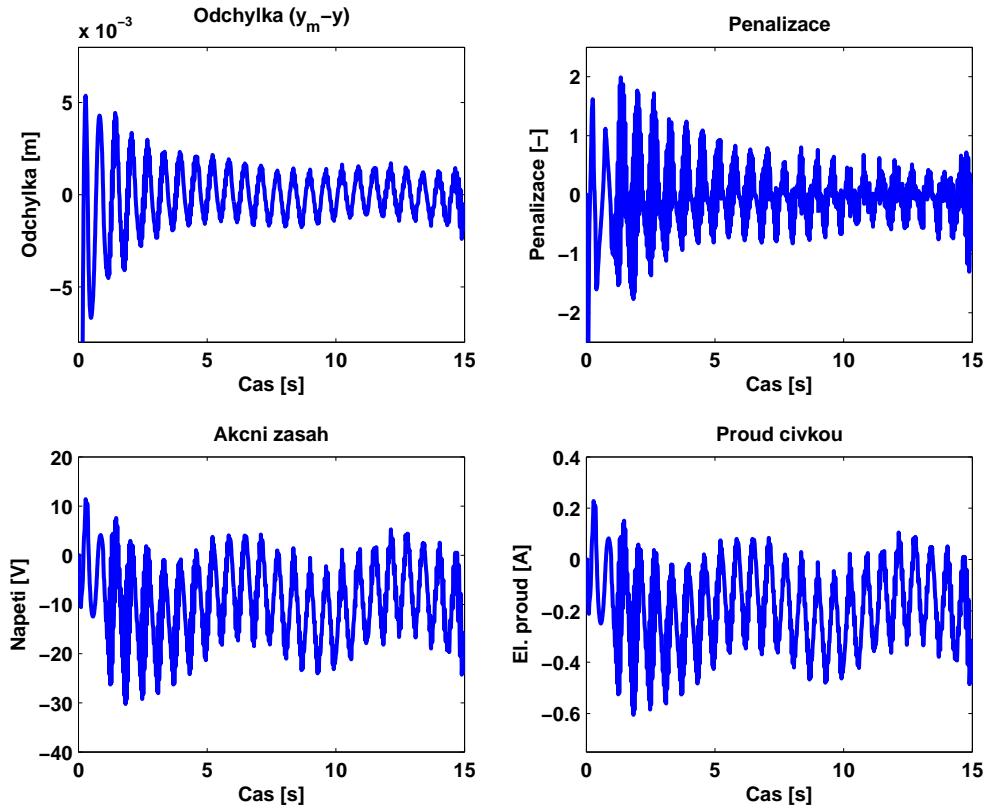
Obrázek 25: Modifikace báze pravidel regulátoru po procesu učení při sledování požadovaného průběhu, zpoždění  $d = 4$



Obrázek 26: Jednotlivé průběhy veličin regulace vztažené k předchozímu obr.25



Obrázek 27: Modifikace báze pravidel regulátoru po procesu učení při sledování požadovaného průběhu, zpoždění  $d = 4$



Obrázek 28: Jednotlivé průběhy veličin regulace vztažené k předchozímu obr.27

### 5.3 Řízení inverzního kyvadla

Řízení inverzního kyvadla je „učebnicová“ úloha řízení, na které byla odzkoušena již celá řada různých metod řízení. Cílem této úlohy je udržovat kyvadlo svisle vzhůru. V případě, že se vozítko pohybuje pouze ve 2 směrech, tak tento systém popisují dvě nelineární diferenciální rovnice. Tento případ je uvažován zde. Často se tyto rovnice linearizují v pracovním bodě, kdy je kyvadlo svislé vzhůru a kde se ho regulací snažíme udržet. Vzhledem k tomu, že FMRLC pracuje s plně nelineárním modelem, nemusíme systém vůbec linearizovat. Předpokládáme dále, že vozík může jezdit po kolejnici, která tvoří kružnici s velkým poloměrem. Z tohoto důvodu odpadá problém s udržením vozítka ve středu kolejnice, jak by to bylo potřeba v případě kolejnice ve tvaru úsečky.

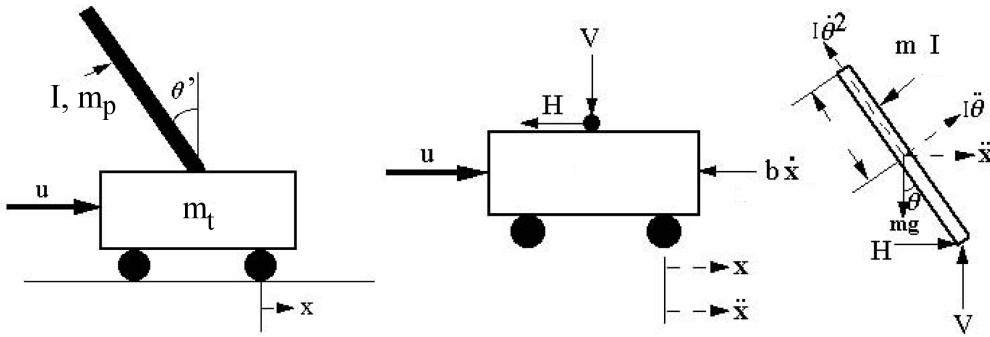
Rovnice pro pohyb inverzního kyvadla lze nalézt ve spoustě publikacích o řízení. V našem příkladě předpokládáme, že kyvadlo zpětně ovlivňuje pohyb vozítka, vozík se tedy nechová jako tvrdý zdroj síly. Výsledné dvě nelineární diferenciální rovnice jsou následující:

$$(I + m_p l) \ddot{\Theta} + m_p g l \sin \Theta = -m_p l \ddot{x} \cos \Theta \quad (25)$$

$$(m_t + m_p) \ddot{x} + b \dot{x} + m_p l \ddot{\Theta} \cos \Theta - m_p l \dot{\Theta}^2 \sin \Theta = u \quad (26)$$

Význam jednotlivých proměnných je následující:

$m_t$ ..hmotnost vozítka,  $m_p$ ..hmotnost kyvadla,  $I$ ..moment setrvačnosti,  $b$ ..koeficient třecí síly,  $u$ ..akční zásah,  $x$ ..pozice vozítka,  $\Theta$ ..úhel kyvadla od svislé dolní polohy



Obrázek 29: Inverzní kyvadlo

Pro simulaci byly zvoleny následující hodnoty:

$$m_t = 0.5 \text{ kg}, m_p = 0.2 \text{ kg}, b = 0.1 \text{ N/ms}, l = 0.3 \text{ m}, I = 0.006 \text{ kg m}^2$$

Po zavedení substituce  $x_1 = \Theta$ ,  $x_2 = \dot{\Theta}$ ,  $x_3 = x$ ,  $x_4 = \dot{x}$  zapíšeme rovnice (25) a (26) ve stavovém popisu:

$$\begin{aligned} \dot{x}_1 &= \dot{\Theta} = x_2 \\ \dot{x}_2 &= \ddot{\Theta} = -\frac{d}{a} \left( \frac{\cos x_1 (au + adx_2^2 \sin x_1 + d^2 g \sin x_1 \cos x_1 - abx_4)}{ac - d^2 \cos x_1} + g \sin x_1 \right) \\ \dot{x}_3 &= \dot{x} = x_4 \\ \dot{x}_4 &= \ddot{x} = \frac{au + adx_2^2 \sin x_1 + d^2 g \sin x_1 \cos x_2 - abx_4}{ac - d^2 \cos^2 x_1}, \end{aligned} \quad (27)$$

kde  $a = I + m_p l$ ,  $c = m_t + m_p$  a  $d = m_p l$ .

### 5.3.1 Výsledky simulací řízení inverzního kyvadla

FMRLC metoda použitá na systému inverzního kyvadla představuje opět řízení nelineárního systému. Narozdíl od předchozích dvou modelů je zde pro srovnání použit i případ, kdy je startovní báze pravidel v regulátoru stejná jako báze v inverzním modelu, která je opět stejná jako u předchozích dvou modelů.

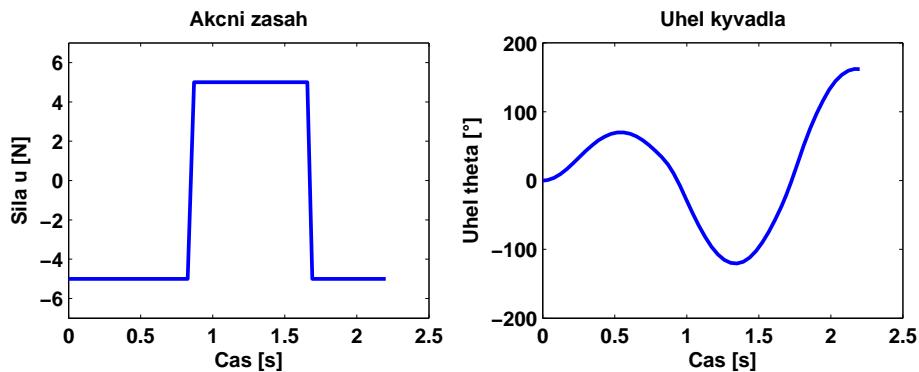
Obr.30 zobrazuje průběh akčního zásahu, kterým kyvadlo přímovazebně vyšvihne do horní polohy, kde ho „převezme“ regulátor. Obr.31 zobrazuje vliv zpoždění  $d$  na adaptaci. Jedná se o případ, kdy je regulátor nastartován bez žádných znalostí, jak systém řídit. Všechny středy výstupních trojúhelníkových funkcí jsou tedy umístěny v nule. Z průběhů je vidět, že opět vychází nejlépe regulace pro zpoždění  $d = 1$ , stejně jako u servosystému. Při zpoždění  $d = 6$  už regulátor nedokázal udržet kyvadlo v horní poloze a kyvadlo spadlo. Následující dva obr.32 a 33 také zobrazují vliv zpoždění  $d$  na adaptaci, ale jedná se už o případ, kdy je startovní báze pravidel shodná s bází pravidel v inverzním modelu. Z tohoto důvodu se mnohem rychleji kyvadlo v horní svislé poloze ustabilizuje. V tomto případě už jsou mezi jednotlivými zpožděními  $d$  malé rozdíly. Nejlepší zpoždění vychází  $d = 2$ , i když rozdíl je nepatrný. Zpoždění  $d > 5$  už způsobují oscilaci kolem žádané hodnoty (viz. obr.33).

Na obr.34 je vidět podobně jako u předchozích modelů výsledná báze pravidel, která vznikla při nulové startovní bázi pravidel. Z obr.35 je vidět, že po 3 s adaptace

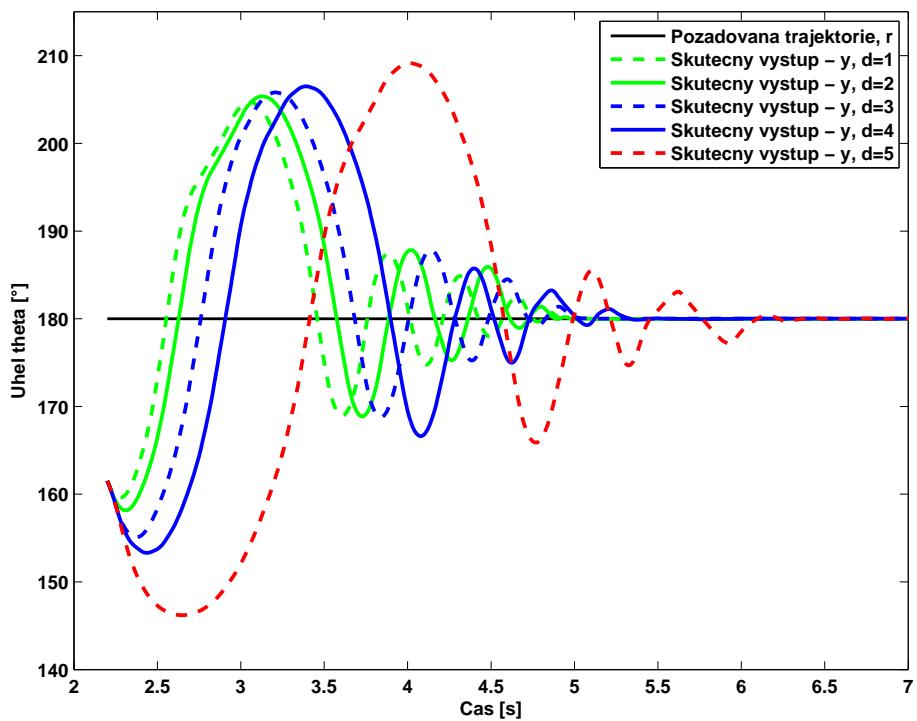
skončí a už není potřeba bázi pravidel více upravovat. Pokud použijeme startovní bázi pravidel shodnou s bází pravidel v inverzním modelu, tak adaptace trvá pouze 1 s (spodní obrázek na obr.37) a není potřeba ji více modifikovat.

Fungování FMRLC algoritmu je velmi závislé mimo jiné i na počtu vstupních funkcí příslušnosti. Vliv počtu vstupních funkcí příslušnosti je na obr.38 a jemu odpovídající si průběhy na obr. 39. Je vidět, že při použití pěti stejných funkcí příslušnosti na vstupu se regulace výrazně zhoršila.

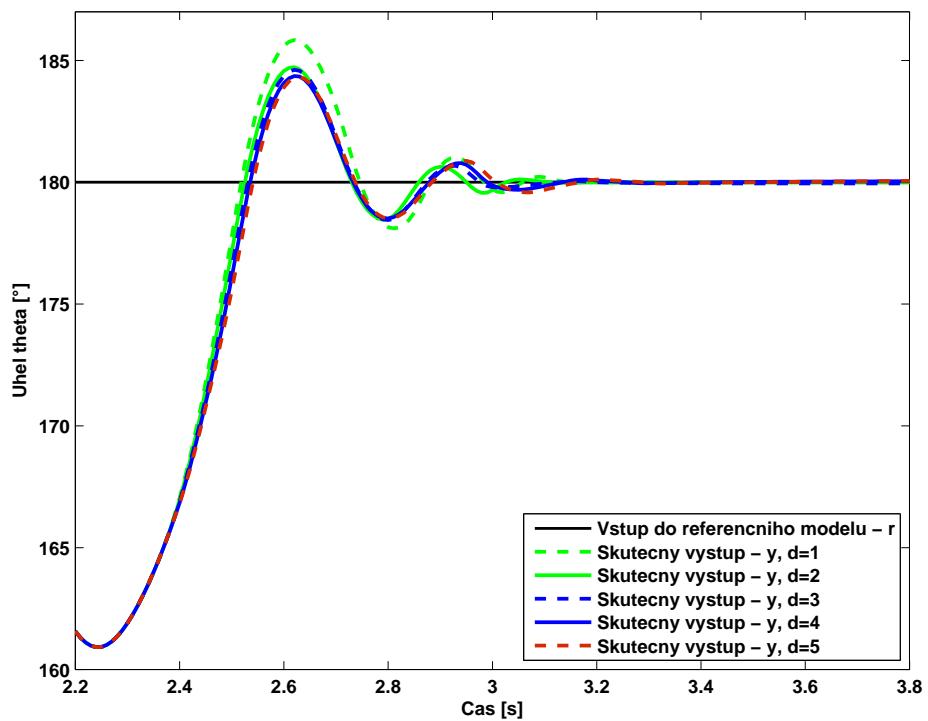
Dále byl vyzkoušen vliv poruch na horní svislou polohu kyvadla. Jak je vidět z obr.40 a 42, tak se regulátor dokáže s vlivem poruch vypořádat dobře, a to jak při nulové startovní bázi pravidel, tak i při startovní bázi pravidel, která je shodná s bází v inverzním modelu.



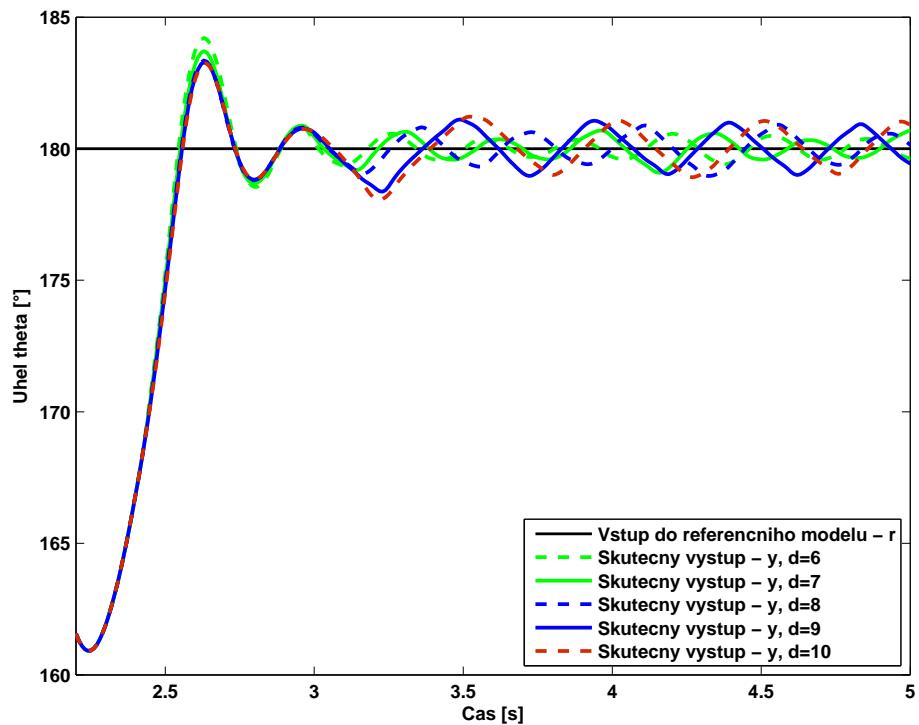
Obrázek 30: Vyšvihnutí kyvadla do horní polohy



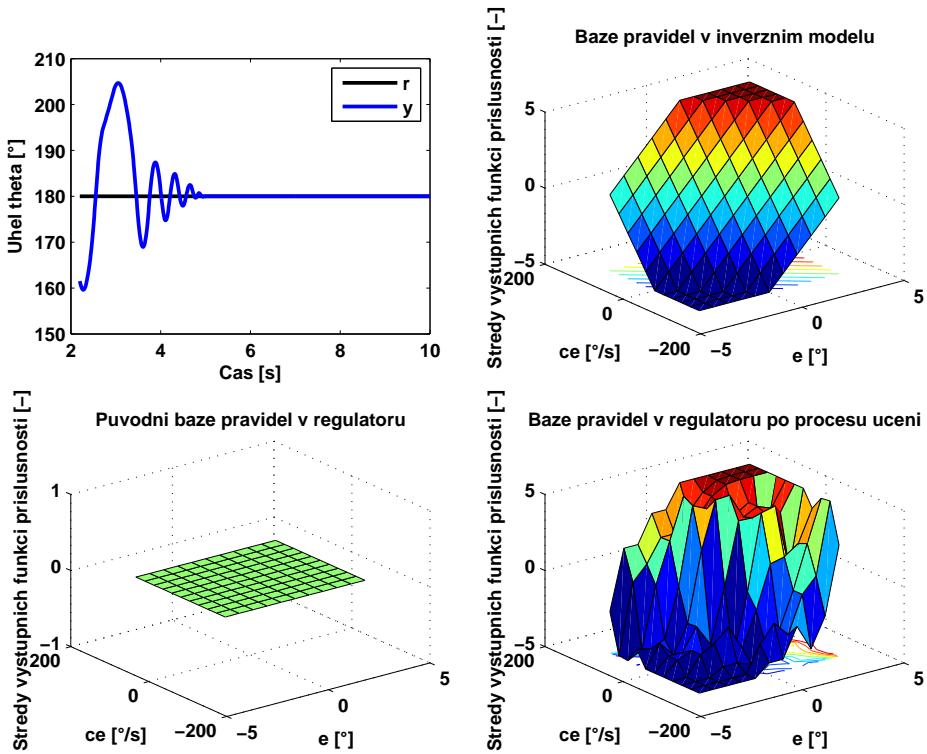
Obrázek 31: Vliv zpoždění  $d = 1, 2, 3, 4, 5$  na nalezení báze pravidel regulátoru (regulátor nastartován s nulovou bází pravidel)



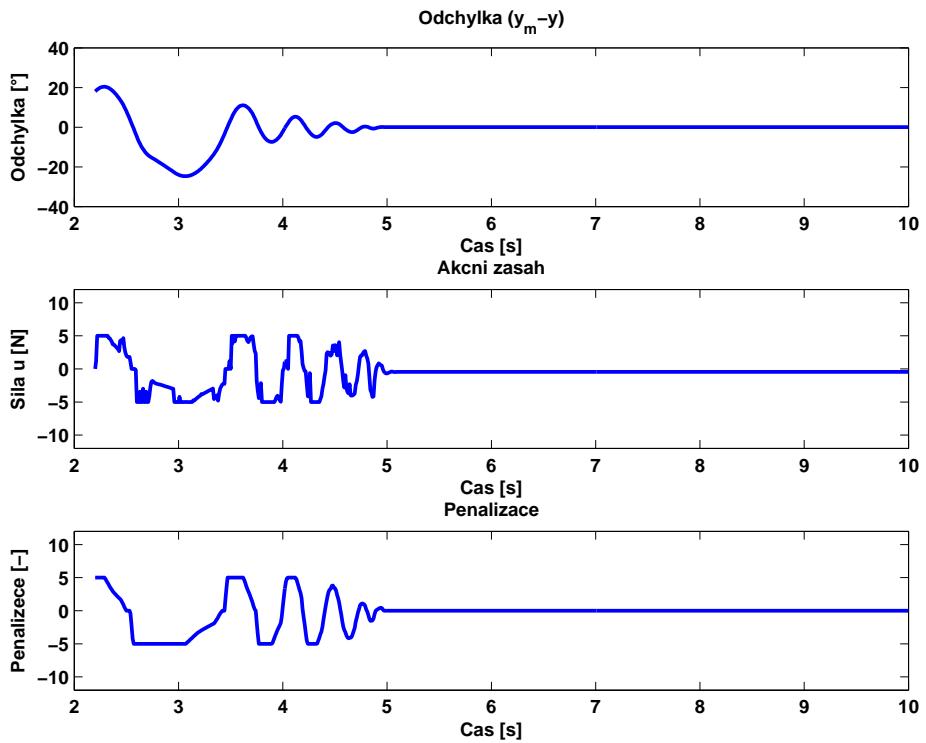
Obrázek 32: Vliv zpoždění  $d = 1, 2, 3, 4, 5$  na regulaci a adaptaci (regulátor nastartován s bází pravidel shodnou v inverzním modelu)



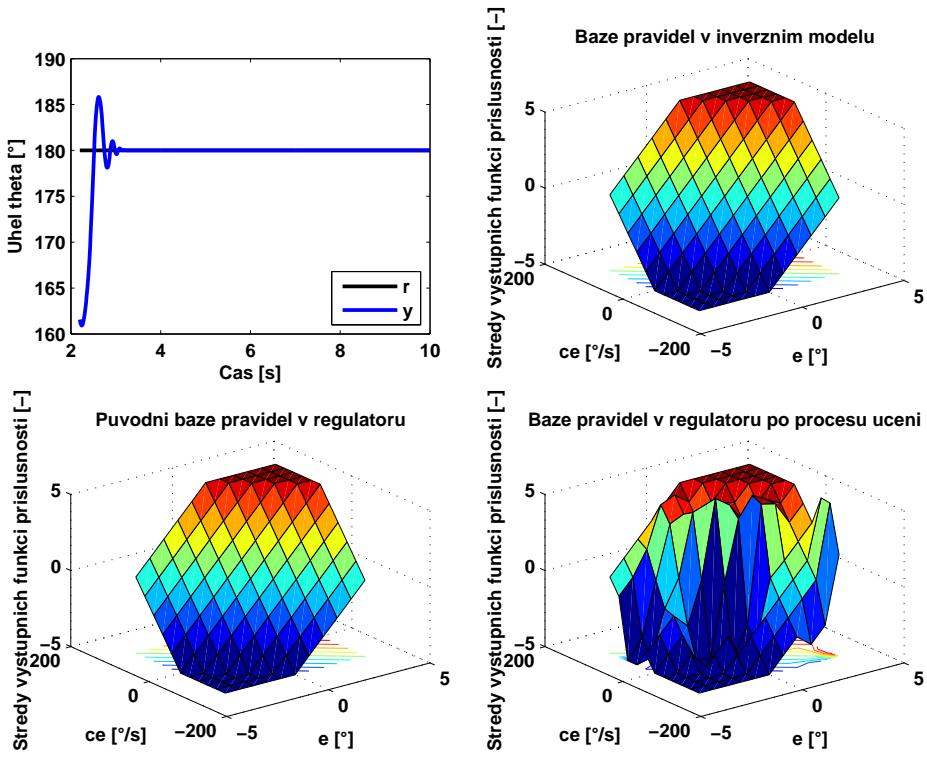
Obrázek 33: Vliv zpoždění  $d = 6, 7, 8, 9, 10$  na regulaci a adaptaci (regulátor nastartován s bází pravidel shodnou v inverzním modelu)



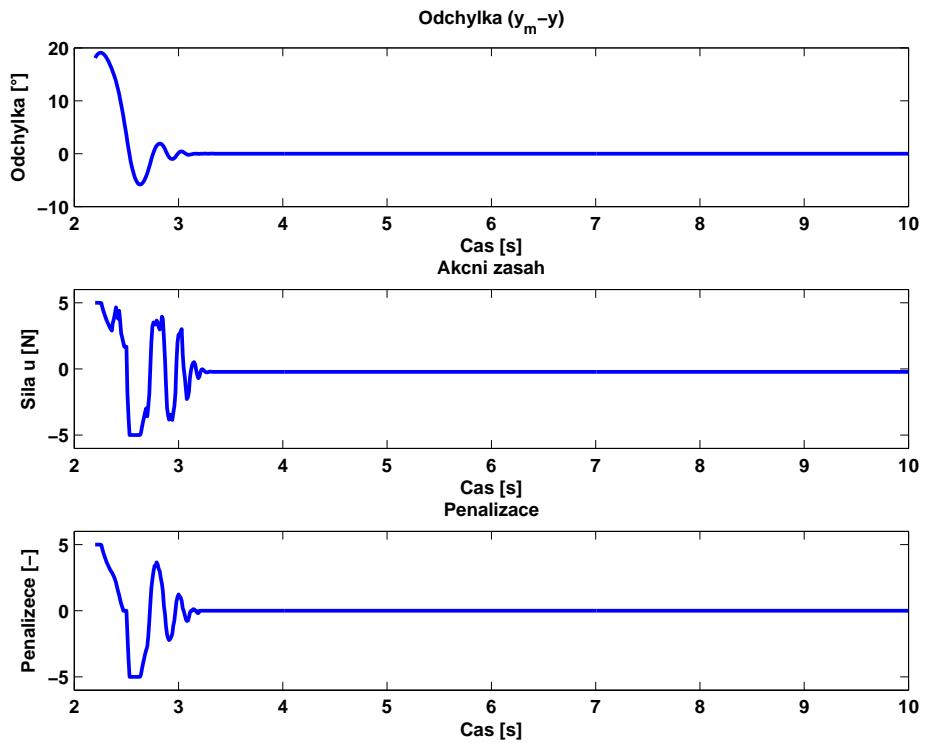
Obrázek 34: Modifikace báze pravidel regulátoru po procesu učení při sledování požadovaného průběhu, zpoždění  $d = 1$



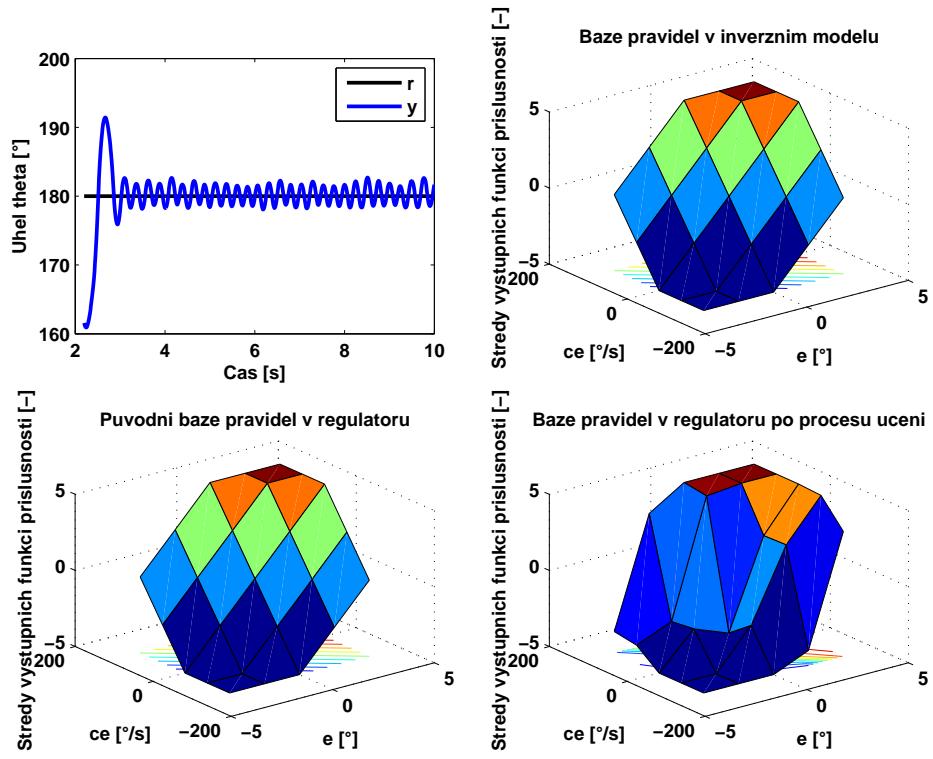
Obrázek 35: Jednotlivé průběhy regulace kyvadla, které jsou vztaženy k předchozímu obr. 34



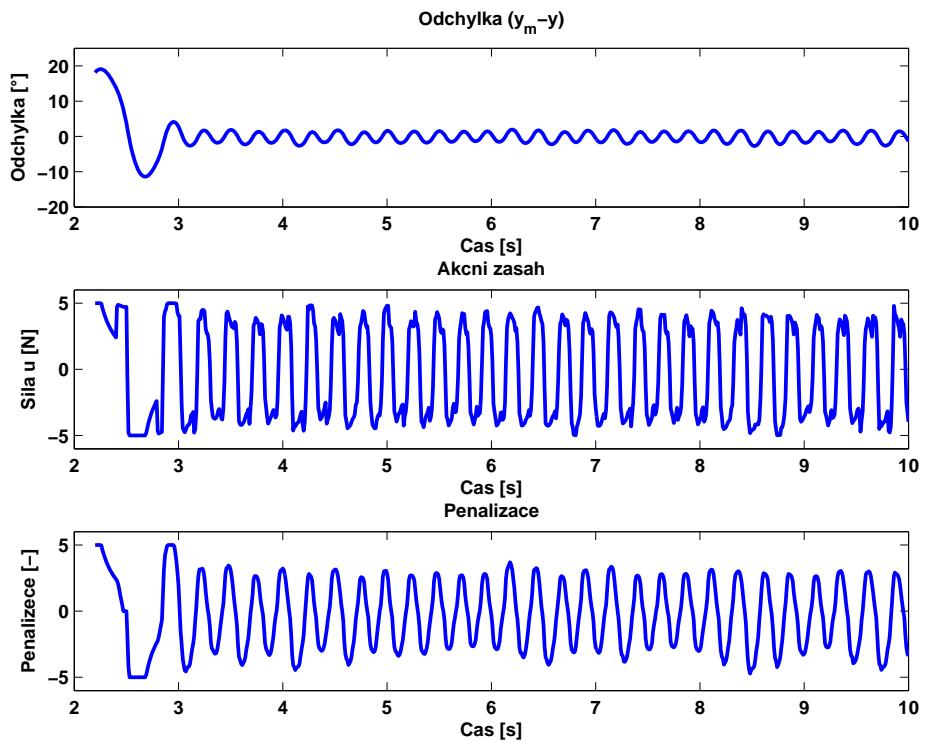
Obrázek 36: Modifikace báze pravidel regulátoru po procesu učení při sledování požadovaného průběhu, zpoždění  $d = 1$



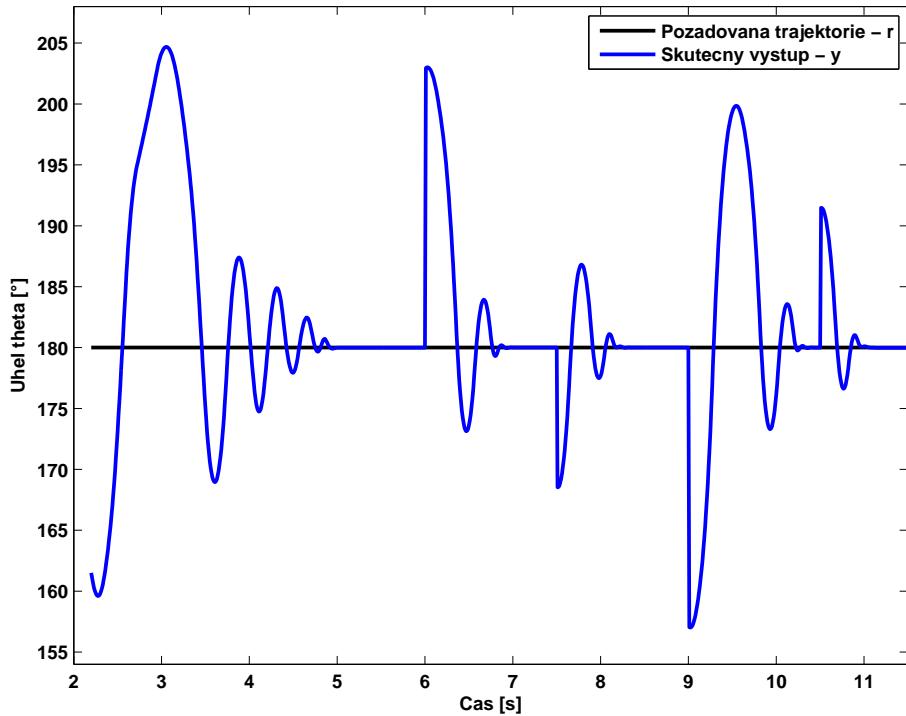
Obrázek 37: Jednotlivé průběhy regulace kyvadla, které jsou vztaženy k předchozímu obr. 36



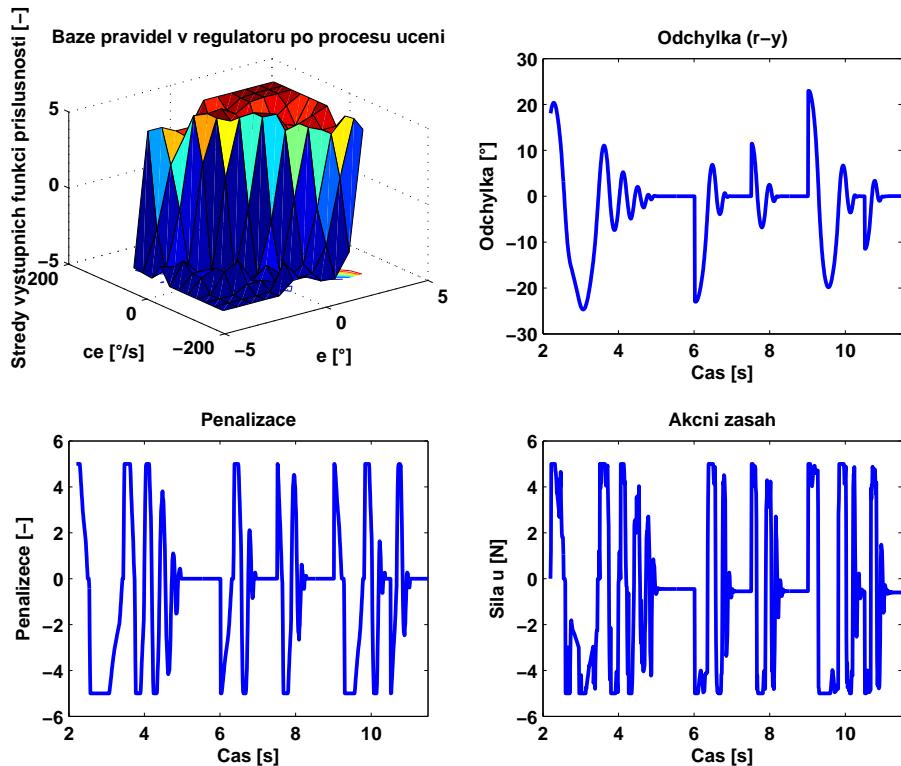
Obrázek 38: Modifikace báze pravidel regulátoru po procesu učení při sledování požadovaného průběhu, 5 trojúhelníkových funkcí příslušnosti na obou vstupech, zpoždění  $d = 1$ , stejné parametry FMRLC shodné z předchozích obrázků



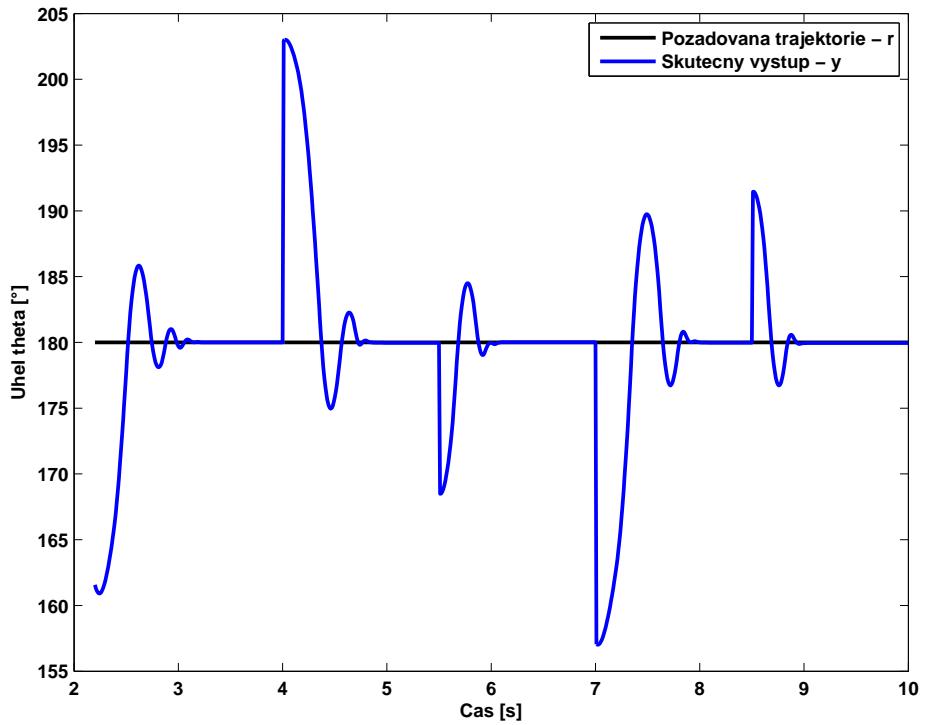
Obrázek 39: Jednotlivé průběhy regulace kyvadla, které jsou vztaženy k předchozímu obr. 38



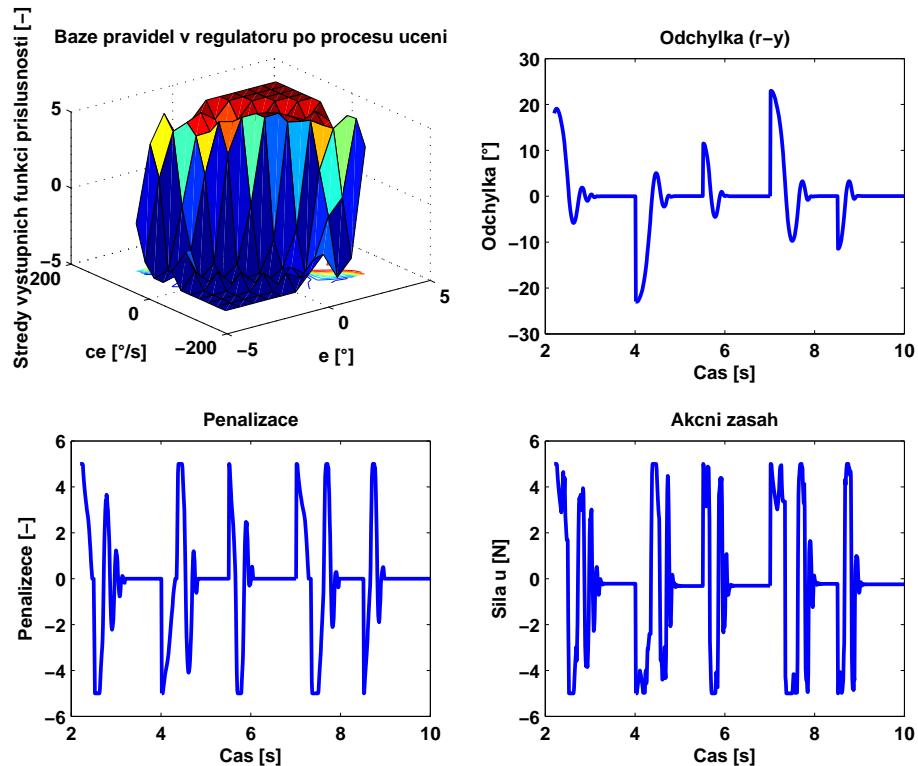
Obrázek 40: Vliv poruch na udržení kyvadla ve svislé poloze, původní báze pravidel je nulová, poruchy: v čase  $t=6$  s je  $23^\circ$ , v čase  $t=7.5$  s je  $-12^\circ$ , v čase  $t=9$  s je  $-23^\circ$ , v čase  $t=10.5$  s je  $12^\circ$



Obrázek 41: Výsledná báze pravidel v regulátoru a průběhy regulace kyvadla, které jsou vztaženy k předchozímu obr. 40



Obrázek 42: Vliv poruch na udržení kyvadla ve svíslé poloze, původní báze pravidel shodná s bází inverzního modelu, poruchy: v čase  $t=4$  s je  $23^\circ$ , v čase  $t=5.5$  s je  $-12^\circ$ , v čase  $t=7$  s je  $-23^\circ$ , v čase  $t=8.5$  s je  $12^\circ$



Obrázek 43: Výsledná báze pravidel v regulátoru a průběhy regulace kyvadla, které jsou vztaženy k předchozímu obr. 42

## 6 Závěr

Schopnosti původního samoorganizujícího se fuzzy regulátoru (poprvé představen v [22]) se ukázaly jako velmi omezené. V praxi takový regulátor je téměř nepoužitelný, a to z několika důvodů. Asi největší nevýhoda je problematické vytvoření penalizační tabulky  $P$ , která má za cíl vhodným způsobem modifikovat bázi pravidel v regulátoru na základě požadovaného průběhu výstupu, který je právě v této tabulce „zakódován“. Neexistuje žádná metoda, která by umožnila pro obecný požadovaný průběh takovou tabulkou vytvořit. Bylo jich sice již několik vytvořeno, ale byly sestaveny do jisté míry metodou *pokus - omyl*. Navíc je regulátor z různých důvodů potřeba navrhnut s jiným počtem pravidel, než s kolika se počítalo při jejich původním sestavování. Transformace těchto tabulek do požadovaných „velikostí“ situaci dále komplikuje. Navíc tabulka  $P$  poskytuje informaci pouze o požadovaném průběhu při sledování jednotkového skoku. Vytvořit tabulku  $P$  pro obecný průběh je skoro nemožné. Nabízí se proto možnost nechat naadaptovat regulátor tak, aby výstup ze systému co nejlépe sledoval jednotkový skok a následně tuto modifikovanou bázi pravidel použít při sledování jiných referenčních průběhů. Tuto situaci zobrazuje obr. 8, ze kterého je ale vidět, že jsme sice použitím modifikované báze pravidel ve srovnání s původní zlepšili sledování požadovaného průběhu, ale i tak není regulace nejlepší. Tyto všechny důvody vedly k tomu, že metoda byla přepracována v metodu již zmíněného adaptivního fuzzy řízení s referenčním modelem na principu učení.

Výhoda FMRLC ve srovnání se samoorganizujícím se regulátorem spočívá hlavně v tom, že jsme zavedením referenčního modelu obešli problematické vytvoření penalizační tabulky v bloku  $P$  v původním samoorganizujícím se regulátoru. Pro všechny tři modely byla použita pouze adaptace, nikoliv učení, a přesto regulace probíhala velmi úspěšně. To je z důvodů, že žádný z těchto modelů nepracoval ve zvlášť různých pracovních podmínkách, které by vyžadovaly navzájem odlišné báze pravidel.

Problémem v obou přístupech se ukázalo nalezení vhodného zpoždění  $d$ . Sice bylo vždy zjištěno na základě simulace, ale skutečnost, že vybrané zpoždění při daném referenčním signálu a stavu systému funguje dobře, ještě neznamená, že bude fungovat i v jiných pracovních oblastech systému. U časově proměnných systémů to může být již velmi vážný problém, protože nalézt nějaký vhodný vztah pro jeho určení je velmi problematické.

Navíc všechny tři systémy měly stejná pravidla, které měly inicializovány v inverzním modelu, a přesto regulace byla úspěšná. V případě ale, že máme k dispozici matematický popis systému, využijeme raději metodu řízení vycházející právě z jeho znalosti, protože FMRLC je pouze heuristická metoda, proto na ni nemůžeme příliš spoléhat. Pro řízení nelineárních systémů, od kterých nepožadujeme vysoce spolehlivé chování, se ukázala jako metoda velice vhodná z důvodu své jednoduchosti. Z tohoto důvodu nemůže být použita v situacích, kde je požadována vysoká spolehlivost. Dále se ukázalo, že u uvedených modelů naprostě vystačí samotná adaptace bez učení. Zapojení s pamětí, a tedy proces učení, se vyplatí spíše u systémů s velmi rozdílnými pracovními podmínkami, kdy by se jednotlivé báze ukládaly právě pro jednotlivé pracovní podmínky. Tento přístup je však hodně aplikacně závislý. Navíc tato metoda není ani v současnosti dostatečně dobře zpracována, takže je tu velký prostor pro její případné vylepšení.

# Reference

## Česká literatura

- [1] Olšák P. and Navara M. *Základy fuzzy množin - skriptum*. Vydavatelství ČVUT, 2007.
- [2] Novák V. *Fuzzy množiny a jejich aplikace*. Nakladatelství technické literatury, 1990.
- [3] Novák V. *Základy fuzzy modelování*. BEN - technická literatura, 2002. ISBN 80-7300-009-1.
- [4] Vysoký P. *Fuzzy řízení*. Vydavatelství ČVUT, 1996. ISBN 80-01-01429-8.
- [5] Hyniová K. *Řídicí technika - přednášky*. Vydavatelství ČVUT, 2006. ISBN 80-01-03368-6.
- [6] Modrlák O. *Fuzzy řízení a regulace - skriptum*. Technická univerzita Liberec, 2004.
- [7] Plšek B. *Umělá inteligence v modelování a řízení*. BEN - technická literatura, 1996. ISBN 80-901984-4-9.
- [8] Pivoňka P. *Optimalizace regulátoru - skriptum*. Fakulta elektrotechniky a komunikačních technologií VUT Brno, 2005.
- [9] Hrachovina M. *Adaptivní fuzzy regulace - diplomová práce*. Technicka univerzita Liberec - Fakulta mechatroniky a mezioborových inženýrských studií, 2003.

## Zahraniční literatura

- [10] Kandel A. and Langholz G. *Fuzzy Control Systems*. CRC-Press, 1993. ISBN 978-0849344961.
- [11] Kovačić B. and Bogdan S. *Fuzzy Controller Design - Theory and Applications*. CRC-Taylor and Francis, zdrojové soubory ke knize - <http://www.fer.hr/predmet/ius>, 2000. ISBN 978-0849337475.
- [12] Driankov D., Hellendoorn H., and Reinfrank M. *An Introduction to Fuzzy Control*. Springer, 1996. ISBN 978-3540606918.
- [13] Driankov D. and Palm R. *Advances in Fuzzy Control*. Studies in Fuzziness and Soft-Computing. Physica-Verlag Heidelberg, 2002. ISBN 978-3790810905.
- [14] Abonyi J. *Fuzzy Model Identification for Control*. Systems and Control - Foundations and Applications. Birkhäuser Boston, zdrojové soubory ke knize - <http://www.fmt.vine.hu/softcomp/book/index.html>, 2003. ISBN 978-0817642389.
- [15] Jantzen J. *Foundations of Fuzzy Control*. Wiley, zdrojové soubory ke knize - <http://www.elektro.dtu.dk/English/education/courses/au/31361/FFC.aspx>, 2007. ISBN 978-0470029633.

- [16] Passino M. K. and Yurkovich S. *Fuzzy Control*. Addison-Wesley, zdrojové soubory ke knize - <ftp://ftp.aw.com/cseng/authors/passino/fc/>, 1998. ISBN 978-0201180749.
- [17] Reznik L. *Fuzzy Controllers Handbook*. Newnes, 1997. ISBN 978-0750634298.
- [18] Wang L.Xi. *A Course In Fuzzy Systems and Control*. Prentice Hall PTR, 1996. ISBN 978-0135408827.
- [19] Antsaklis J. P. and Passino K M. *An Introduction to Intelligent and Autonomous Control*. Kluwer Academic Publisher, 1992. ISBN 978-0792392675.
- [20] Spooner J. T., Maggiore M., Ordonez R., and Passino M. K. *Stable Adaptive Control and Estimation for Nonlinear Systems - Neural and Fuzzy Approximator Techniques*. Wiley, zdrojové soubory ke knize - [http://www.ece.osu.edu/~passino/SACENSbooksoftware/file\\_index.html](http://www.ece.osu.edu/~passino/SACENSbooksoftware/file_index.html), 2002. ISBN 978-0471415466.
- [21] Wang L. X. *Adaptive Fuzzy Systems and Control - Design and Stability Analysis*. Prentice Hall, 1994. ISBN 978-0130996312.

## Články z časopisů a konferenční příspěvky

- [22] Mamdani E. H. and Procyk T.J. A linguistic self-organizing process controller. *Automatica*, 15:15–30, 1979. Pergamon Press.
- [23] Yamazaki T. An improved algorithm for a self-organizing controller and its experimental analysis. *PhD thesis*, 1982. London University.

## Literatura k programovému vybavení

- [24] Sivanandam S. N., Sumathi S., and Deepa S. N. *Introduction to Fuzzy Logic using MATLAB*. Springer, 1997. ISBN 978-3540357803.