



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Soft computingové metody identifikace a regulace

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 3902T005 – Automatické řízení a inženýrská informatika
Autor práce: **Bc. Václav Stražil**
Vedoucí práce: Ing. Tomáš Náhlovský, Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Václav Stražil**
Osobní číslo: **M14000190**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Automatické řízení a inženýrská informatika**
Název tématu: **Soft computingové metody identifikace a regulace**
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s metodami a popište princip fungování neuronových sítí v oblasti identifikace systémů.
2. Demonstrujte fungování neuronových sítí při identifikaci vybraných přenosových funkcí.
3. Analyzujte možnosti voleb jednotlivých parametrů sítě - vstupní signál, počet neuronů a vrstev, omezení neuronových sítí.
4. Ukažte vlastnosti a možnosti neuronové sítě při identifikaci reálného procesu.
5. Na zvolené úloze realizujte některou formu inteligentního řízení (fuzzy).

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **40–50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] **VONDRÁK, Ivo. Umělá inteligence a neuronové sítě. Dot. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 1998, 139 s. ISBN 80-7078-259-5.**
- [2] **Fausett, L. V.: Fundamentals of Neural Networks. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 1994.**

Vedoucí diplomové práce: **Ing. Tomáš Náhlovský, Ph.D.**

Ústav mechatroniky a technické informatiky

Konzultant diplomové práce: **Ing. Lukáš Hubka, Ph.D.**


Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: **10. října 2016**

Termín odevzdání diplomové práce: **15. května 2017**


prof. Ing. Zdeněk Pliva, Ph.D.
děkan




doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2016

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Tímto bych rád poděkoval vedoucímu práce, Ing. Tomášovi Náhlovskému Ph.D., za vedení práce, rady a věcné připomínky. Dále kolegům ze zaměstnání, zejména pak Dipl.-Ing. Jiřímu Konvalinkovi za vstřícný přístup a trpělivost. V neposlední řadě potom rodině za podporu ve studiu.

Abstrakt

Tato práce se zabývá využitím softcomputingových metod v oblasti regulace a identifikace dynamických soustav. Popisuje základy fuzzy množin, fuzzy regulace, možnosti aplikace fuzzy regulace a porovnává výhody a nevýhody fuzzy regulace oproti konvenčnímu přístupu. Dále pojednává o identifikaci dynamických soustav pomocí neuronových sítí. Je zde uveden biologický základ myšlenky, jeho počítačová implementace a základní struktura zapojení neuronové sítě, vhodná pro identifikaci systémů. Kromě praktického použití neuronových sítí, nejen v oblasti identifikace, je zde diskutována volba struktury a parametrů neuronových sítí a vhodnost užití identifikace pomocí neuronových sítí na různých typech dynamických soustav. Dále je zde představeno několik laboratorních modelů, na kterých byly znalosti regulace a identifikace softcomputingovými metodami aplikovány prakticky.

Klíčová slova

softcomputing, regulace, fuzzy, identifikace, neuronové sítě

Abstract

This work deals with the use of the softcomputing methods for control and dynamic system identification. It describes the fundamentals of fuzzy sets, fuzzy control, fuzzy applications and compares the advantages and disadvantages of fuzzy control compared to the conventional approach. It also deals with the dynamical system identification using neural networks. There are shown biological basis thoughts, computer implementation and a neural network basic structure suitable for identification systems. In addition to the practical use of neural networks, not only in the field of identification, there is a discussion about neural networks structure options and parameters and the appropriateness of the use of identification using neural networks for different types of dynamic systems. Control and identification knowledge of softcomputing methods were applied practically on several laboratory models.

Keywords

softcomputing, control, fuzzy, identification, neural networks

Obsah

ÚVOD.....	9
1. FUZZY ŘÍZENÍ.....	10
1.1. Fuzzy množiny.....	10
1.2. Fuzzy regulace.....	12
1.3. Aplikace fuzzy regulace, fuzzy množin.....	15
1.3.1 Metro.....	15
1.3.2 Jeřáb.....	16
1.3.3 Systém ABS.....	16
1.3.4 Další použití.....	16
1.4. Fuzzy SW a HW prostředky.....	17
2. NEURONOVÉ SÍTĚ.....	18
2.1. Biologický základ.....	18
2.2. Model neuronu.....	19
2.3. Neuronové sítě.....	19
2.4. Aplikace neuronových sítí.....	21
2.4.1 Zpracování obrazu.....	21
2.4.2 Chemický a farmaceutický výzkum.....	21
2.4.3 ALVINN, Autonomně řízené vozidlo.....	21
2.4.4 Další použití.....	22
2.5. Identifikace dynamických systémů.....	22
2.5.1 Postup modelování systému pomocí neuronové sítě.....	22
2.5.2 SISO systém.....	24
2.5.3 MIMO systém.....	27
2.5.4 Systém se saturací.....	31
2.5.5 Astatický systém.....	33
2.5.6 Nelineární systém.....	35
2.5.7 Shrnutí.....	37
3. POPIS FYZICKÝCH MODELŮ.....	38
3.1. DC motor.....	38
3.2. Fén.....	38
3.3. Vrtulník CE 150.....	39
4. PRAKTICKÁ ČÁST.....	41
4.1. Identifikace soustavy – fén.....	41
4.2. Identifikace a řízení soustavy – motor.....	42
4.1. Identifikace a řízení soustavy – vrtulník.....	48
5. ZÁVĚR.....	55
POUŽITÁ LITERATURA.....	57
PŘÍLOHY.....	60

Seznam obrázků

1-1 Porovnání klasické teorie množin a teorie fuzzy množin na příkladu výšky člověka	10
1-2 Grafické zobrazení základních operací s fuzzy množinami	11
1-3 Regulační řetězec s fuzzy regulátorem	12
1-4 Vstupní funkce příslušnosti	13
1-5 Výstupní funkce příslušnosti	13
1-6 Příklad výstupu z fuzzy regulátoru	14
1-7 Jedna ze základních struktur fuzzy regulátoru	15
2-1 Popis neuronu. Zdroj: [39]	18
2-2 Počítačový model neuronu. Zdroj: [41].....	19
2-3 Programový model neuronové sítě.	20
2-4 Náhled na NARX síť v softwaru MATLAB.	20
2-5 Odezva SISO systému na jednotkový pulz	24
2-6 Identifikační signál a odezva SISO systému	25
2-7 Kontrolní signál a odezva SISO systému	25
2-8 Porovnání dynamické soustavy a modelu	26
2-9 Přesnost neuronové sítě v závislosti na parametrech	26
2-10 Odezva MIMO systému na jednotkový pulz.....	28
2-11 Identifikační signál pro MIMO soustavu	28
2-12 Kontrolní signál použitý na MIMO systém.....	29
2-13 Porovnání MIMO systému a modelu	29
2-14 Teoretický a reálný průběh přeučení neuronové sítě. Zdroj [35].	30
2-15 Závislost přesnosti sítě na počtu epoch	30
2-16 Síť s největší přesností (dle NRMSE)	31
2-17 Identifikační signál systému se saturací	31
2-18 Kontrolní signál, odezva systému na kontrolní signál.....	32
2-19 Porovnání modelu a odezvy soustavy	32
2-20 Identifikační signál astatického systému	33
2-21 Kontrolní signál astatického systému	34
2-22 Porovnání soustavy a neuronové sítě u astatické soustavy	34
2-23 Závislost učení na délce identifikačního signálu.....	35
2-24 Použitý nelineární systém.....	35

2-25	Identifikační signál a odezva nelineárního systému	36
2-26	Kontrolní signál a odezva nelineárního systému	36
2-27	Porovnání modelu a odezvy soustavy	36
3-1	Model stejnosměrného motoru ve školní laboratoři	38
3-2	Laboratorní model fěnu s teplotním čidlem	39
3-3	Vrtulník CE 150. Zdroj:[38]	40
4-1	Identifikační signál a odezva systému pro laboratorní model fěnu	41
4-2	Kontrolní signál a odezva systému modelu fěnu	42
4-3	Porovnání fyzické soustavy motoru s modelem	42
4-4	Identifikační signál a odezva systému pro model motoru a tachodynamu	43
4-5	Kontrolní signál a odezva systému na model motoru a tachodynamu	43
4-6	Výsledek učení neuronové sítě na úloze motoru	43
4-7	Schéma zapojení pro regulaci na modelu vytvořeném neuronovou sítí	44
4-8	Návrh regulátoru na simulačním modelu	45
4-9	Funkce příslušnost fuzzy PI regulátoru	45
4-10	Porovnání regulace PI regulátoru a fuzzy PI regulátoru na fyzickém systému	46
4-11	Funkce příslušnosti fuzzy PID regulátoru	47
4-12	Porovnání regulace PID regulátoru a fuzzy PID regulátoru na fyzické soustavě ..	47
4-13	Identifikační signál pro model vrtulníku	48
4-14	Kontrolní signál pro model vrtulníku	48
4-15	Identifikovaný model vrtulníku	49
4-16	Navrhnutý regulátor na simulačním modelu a ověření funkčnosti na fyzickém systému	50
4-17	Funkce příslušnosti fuzzy regulátoru pro hlavní motor modelu	51
4-18	Fuzzy regulace v uzavřené smyčce	52
4-19	Akční zásah fuzzy regulátoru	53
4-20	Regulace v uzavřené smyčce s vazbou od zadního motoru	53
4-21	Porovnání fuzzy regulátoru a PID regulátoru	54
4-22	Porovnání akční veličiny fuzzy regulátoru a akční veličiny PID regulátoru	54

Úvod

Fuzzy logika, neuronové sítě, evoluční algoritmy a další odvětví softcomputingu jsou poměrně nové přístupy k strojovému řešení problémů. Všechny však spojuje inspirace v přírodních jevech. U fuzzy logiky je to lidské chápání problému (především pak slovní popis), u neuronových sítí je to princip funkce mozku, a nakonec u evolučních algoritmů je to, jak název napovídá, teorie evoluce. Tato práce se zabývá především regulací v uzavřené smyčce pomocí fuzzy regulátoru a identifikaci systémů prostřednictvím neuronových sítí.

Přesto, že první regulátory se objevily již více než 200 lety (Wattův odstředivý regulátor), jedná se stále o aktuální problematiku. S rozvojem elektroniky, průmyslu a techniky obecně stále roste snaha o přesnější, komfortnější, spolehlivější, bezpečnější procesy. Mimo klasické teorie regulace, založené na PID regulátorech, tak vznikají stále nové přístupy, jak takových technologických pochodů dosáhnout. Mezi tyto se řadí prediktivní řízení, robustní řízení nebo právě fuzzy regulace. V této práci je možné se dočíst o matematickém základu fuzzy množin, principech fuzzy regulace, softwarových prostředcích pro implementaci i praktické využití na laboratorních modelech.

Úloha identifikace slouží k vytvoření modelu systému s co možná nejvěrnější podobou chování a vlastností. Úspěšně provedená identifikace tak vede na model, na kterém lze testovat odezvu systému na různé situace, případně dokonce reakci soustavy předvídat. Použití neuronových sítí na takovou funkci je také předmětem této práce.

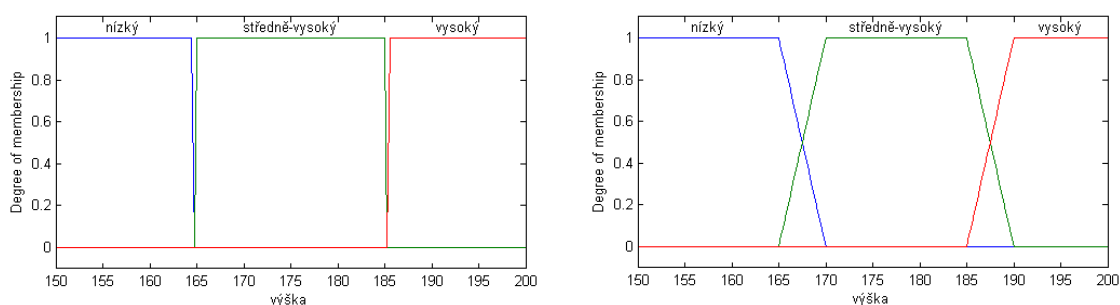
V celé práci je používáno spojení neuronové sítě, není-li řečeno jinak, pro umělé neuronové sítě. Práce předpokládá alespoň základní znalosti z oboru regulačních obvodů a dynamických soustav.

1. Fuzzy řízení

K pochopení principu fuzzy regulace je nutné znát základy teorie fuzzy množin. O teorii fuzzy množin i o fuzzy regulaci byla již publikována spousta literatury, a to jak české, tak cizojazyčné. O této problematice je zde zmíněn jen stručný základ, potřebný pro pochopení myšlenky fuzzy řízení a k zavedení pojmů potřebných pro další výklad. Hlubší teoretické znalosti lze nalézt např. v [1],[2].

1. 1. Fuzzy množiny

V klasické teorii množin (založené na binární logice) prvek do množiny patří nebo nepatří. Takovou množinu nazýváme ostrou. Fuzzy množiny jsou neostře, prvek může patřit do dané množiny, nepatřit do množiny nebo částečně patřit do množiny. Tato skutečnost bude demonstrována na příkladu. Řekněme, že bude rozhodnuto, kdy nazveme člověka vysokým. V klasické teorii množin je nutné definovat přesnou hranici, kdy bude člověk nazván vysokým, například 185 centimetrů. To by ale znamenalo, že člověk, měřící 184,9 cm již vysoký není. Nyní bude tento příklad ukázán ve fuzzy množinách.



1-1 Porovnání klasické teorie množin a teorie fuzzy množin na příkladu výšky člověka

Řekněme, že člověk měří 180 centimetrů. Podle obrázku 1-1, tak nepatří tento člověk do množiny vysoký jak v klasické teorii množin, tak ani ve fuzzy množinách. Člověk měřící 190 centimetrů patří v obou případech do množiny vysoký. Rozdíl nastává, pokud člověk měří 187,5 centimetru. Ve fuzzy množinách říkáme, že tento člověk je vysoký se stupněm příslušnosti 0,5. Se stejným stupněm příslušnosti patří i do množiny středně-vysoký. Proces přiřazení hodnot do fuzzy množin se nazývá fuzzyfikace.

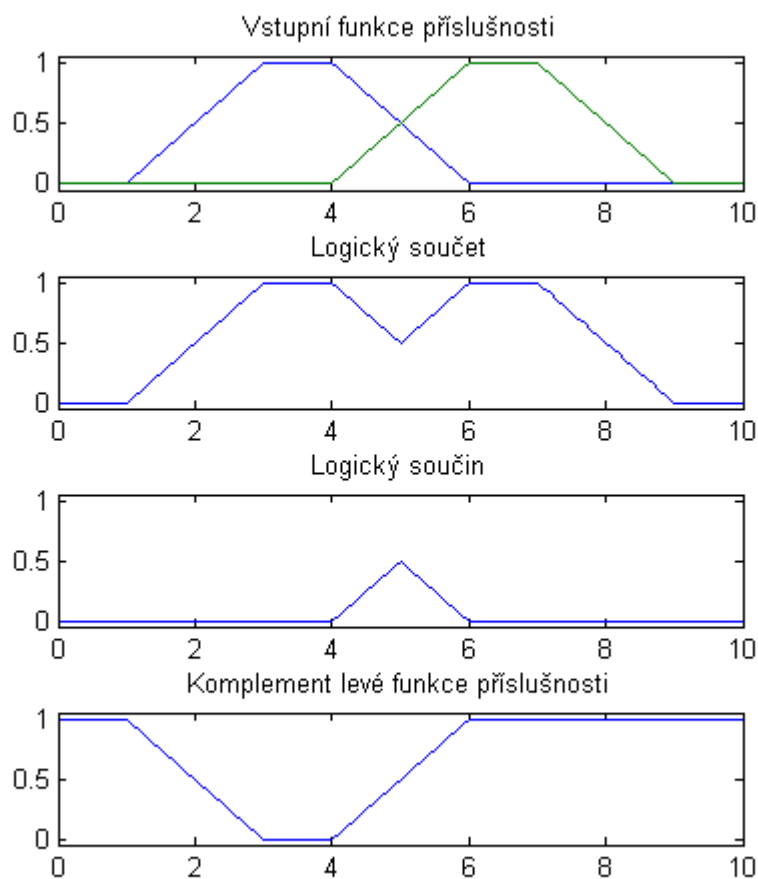
Funkce, zobrazující velikost příslušnosti hodnoty nazýváme funkcí příslušnosti. Tato funkce může mít libovolný tvar (pokud splňuje definici funkce) a musí mít obor

hodnot v intervalu $\langle 0;1 \rangle$. Definiční obor nazveme univerzum. Jev, který popisujeme nazveme lingvistickou proměnnou. V příkladu výše je tedy lingvistická proměnná výška člověka popsána třemi lingvistickými hodnotami {nízký, středně-vysoký, vysoký}.

Nyní bude definováno několik operací s fuzzy množinami. Jedná se o logický součin, logický součet a komplement množiny. V následující tabulce jsou pro ilustraci provedeny tyto operace na vstupních lingvistických hodnotách $mf1$ a $mf2$.

Tabulka 1 Logické operace s fuzzy množinami

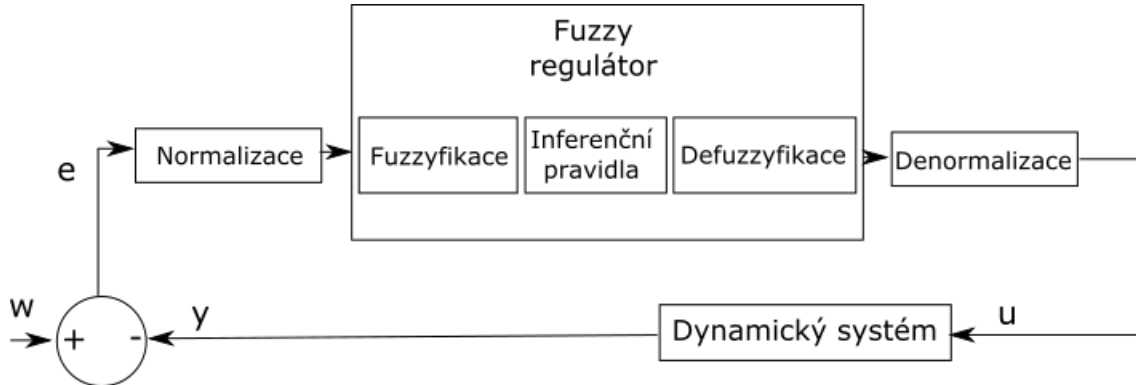
Logický součet (OR)	$mf3 = \max \{mf1, mf2\}$
Logický součin (AND)	$mf4 = \min \{mf1, mf2\}$
Komplement (NOT)	$mf5 = 1 - mf1$



1-2 Grafické zobrazení základních operací s fuzzy množinami

1. 2. Fuzzy regulace

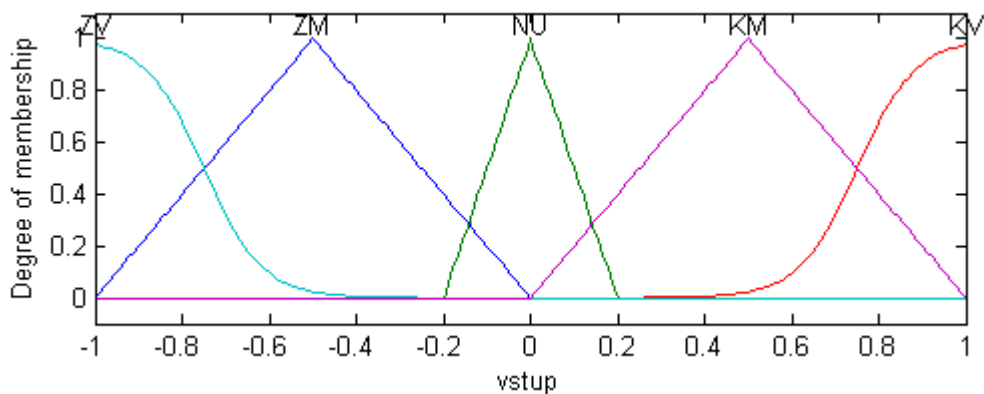
Na obrázku 1-3 je zobrazen regulační řetězec s fuzzy regulátorem. Symbol „ u “ značí akční veličinu, tedy zásah regulátoru do regulované soustavy, „ y “ je regulovanou veličinou, „ w “ je žádaná hodnota. rozdíl „ $e=w-y$ “ nazýváme regulační odchylkou.



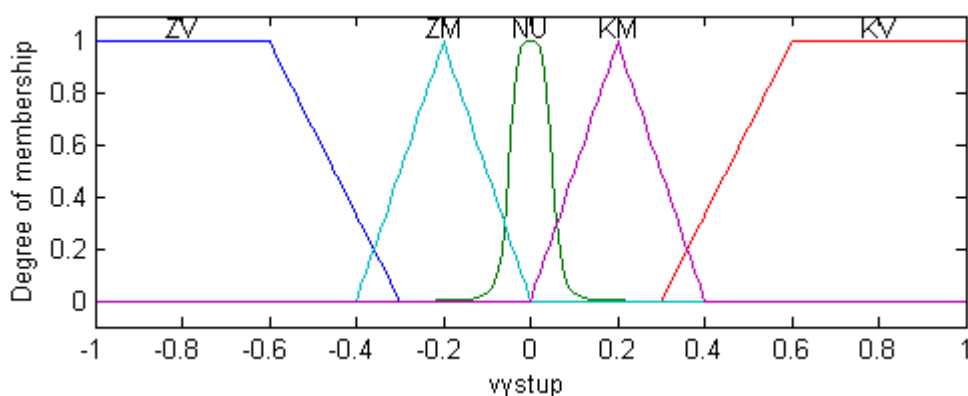
1-3 Regulační řetězec s fuzzy regulátorem

Blok fuzzyfikace převádí vstupní fyzikální veličinu (regulační odchylku) na lingvistickou vstupní veličinu. Pomocí inferenčních pravidel je této vstupní veličině (popřípadě veličinám) přiřazena výstupní lingvistická veličina (veličiny). Výstupní lingvistická veličina je poté převedena zpět na výstupní fyzikální veličinu. Poslední zmíněný proces se nazývá defuzzyfikace. Bloky normalizace a denormalizace zajišťují případné převedení na správný rozsah hodnot. Pro lepší představu bude myšlenka fuzzy regulace demonstrována na příkladu.

Nejdříve budou vytvořeny funkce příslušnosti pro vstup do fuzzy regulátoru (v tomto příkladu regulační odchylku). Skládá se z pěti funkcí příslušnosti (obr. 1-4), vyjadřující velikost regulační odchylky. Je to záporná velká (ZV), záporná malá (ZM), nulová (NU), kladná malá (KM), kladná velká (KV). Podobně vytvoříme i výstupní funkci. Na příkladu lze vidět, že je možné kombinovat různé tvary funkcí příslušnosti.



1-4 Vstupní funkce příslušnosti



1-5 Výstupní funkce příslušnosti

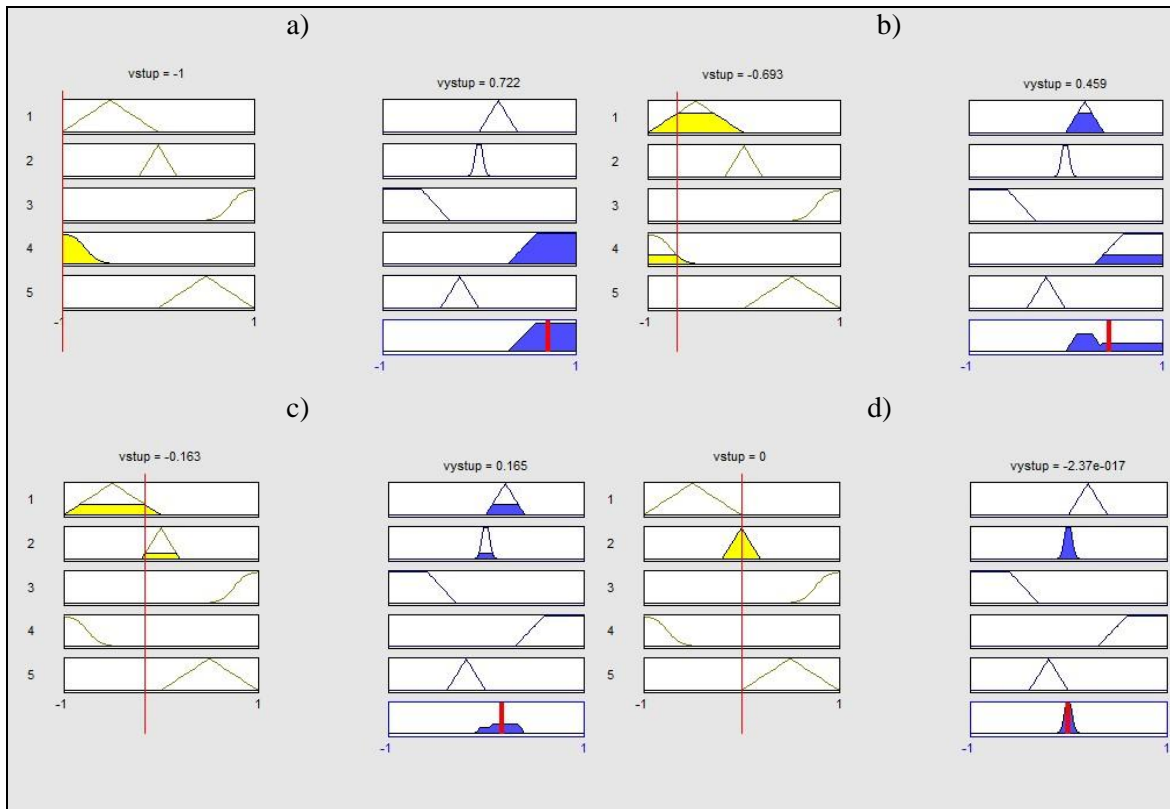
Nyní bude vytvořena sada inferenčních pravidel, které přiřazují vstupním funkcím funkce výstupní. Pokud je více vstupních proměnných, je možné přiřadit funkci výstupní pomocí operace nad fuzzy množinami (AND, OR, NOT). Inferenční pravidla jsou vytvořena na základě expertních znalostí, či obecných metapravidel. Popsaných například v [2].

1. If (vstup is ZM) then (vystup is KM)
2. If (vstup is NU) then (vystup is NU)
3. If (vstup is KV) then (vystup is ZV)
4. If (vstup is ZV) then (vystup is KV)
5. If (vstup is KM) then (vystup is ZM)

Tím je vytvořen jednoduchý fuzzy P regulátor. Dále bylo zvoleno, že regulátor je typu Mamdani (viz např. [8]) a defuzzyfikace probíhá metodou těžiště. Na obrázku 1-6 je několik ukázek, jak celý proces probíhá. Na obrázku a) je vstupní hodnota záporná velká se stupněm příslušnosti 1. Vstupní hodnotě záporná velká je přiřazeno pravidlo takové, že pokud je vstupní hodnota záporná velká, výstupem bude hodnota kladná velká. Jelikož je vstupní hodnota se stupněm příslušnosti 1, odpovídá výstupní hodnotě

plocha pod celou funkcí příslušnosti. Pro zvolenou defuzzifikační metodu (metoda těžiště plochy) je výsledek na hodnotě 0,722.

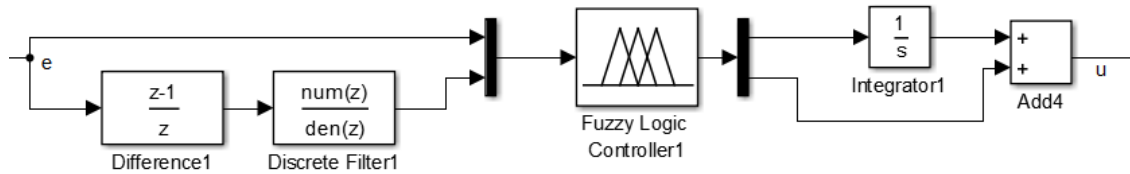
Na obrázku b) a c) je vidět, že vstupní hodnota přísluší dvěma funkcím příslušnosti, tedy i výstup se bude skládat ze dvou ploch. Na obrázku d) je regulační odchylka nulová, patří tedy vstupní funkci příslušnosti NU se stupněm příslušnosti 1. To podle inferenčních pravidel odpovídá výstupu NU. Těžiště této plochy je v bodě 0.



1-6 Příklad výstupu z fuzzy regulátoru

Již z tohoto jednoduchého příkladu lze odvodit několik obecných „pravidel“, jak navrhnout funkce příslušnosti pro účely regulace. Kupříkladu pokud by byla výstupní funkce příslušnosti NU nesymetrická, pro vstupní hodnotu 0 by těžiště neleželo v nule. Nebo pokud by v těsné blízkosti vstupní hodnoty nula nebyla jedna z funkcí KM a ZM, zásah regulátoru na malou odchylku by byl vždy nulový.

Výše navržený regulátor obsahuje jen proporcionální složku, nedošlo by tedy k odstranění trvalé regulační odchylky, protože byla-li by regulační odchylka nulová i výstup by byl nulový, došlo by znovu k rozvážení stavu. Proto by fuzzy regulátor měl obsahovat integrační složku. Pro některé aplikace je vhodné také vytvořit jako další vstup do regulátoru derivaci z regulační odchylky. Kombinací všech těchto složek pak vznikne fuzzy PID regulátor.



1-7 Jedna ze základních struktur fuzzy regulátoru

1. 3. Aplikace fuzzy regulace, fuzzy množin

Fuzzy regulátory se často využívají tam, kde je regulovaná soustava nelineární, popřípadě proměnlivá (neurčitá) nebo pro MIMO (Multiple Input Multiple Output) systémy. Hlavní myšlenkou fuzzy regulace je fakt, že pro řízení procesu není nutné znát matematicko-fyzikální model, ale využívá se znalostí tzv. experta, který zná chování soustavy a dokáže ho slovně popsat. Podobně jako řidič auta nezná matematicko-fyzikální model auta, ale ví, jak se auto zachová. Jedno z pravidel vyřčených expertem by pak mohlo například znít: Jedu-li po rovině rychlostí povolenou ve městě, mám zařazený čtvrtý rychlostní stupeň a lehce stlačím pedál plynu, rychlost auta se nebude měnit. V této kapitole bude uvedeno několik příkladů, kde byli fuzzy regulátory úspěšně nasazeny.

1. 3. 1 Metro

Jedno ze nejznámějších použití fuzzy regulátorů je řízení rozjezdu a brždění soupravy metra v japonském městě Sendai[3], [6], [7]. Jedná se o první použití fuzzy logiky v železniční dopravě. Realizaci provedla firma Hitachi, provoz této necelých 15 kilometrů dlouhé dráhy byl spuštěn roku 1987.

Vstupem do regulátoru je rychlost, signály z pozičních značek, umístěných ve stanici, signály ze supervizní stanice. Samotný chod regulátoru byl rozdělen do dvou částí. První je řízení rozjezdu soupravy a udržování konstantní rychlosti mezi stanicemi a druhou je zastavení v cílové stanici. V obou případech byl kladen důraz na komfort a bezpečnost. Systém dosáhl velice hladkých rozjezdů a zastavení, zajišťující pohodlnou jízdu i pro stojící cestující.

Bylo provedeno i porovnání fuzzy přístupu s konvenčním PID a s řízením strojevodcem. Zjistilo se, že PID regulátor velmi často měnil svoji hodnotu, což způsobovalo menší komfort. Naopak člověk nezasahoval tak často, o to jeho zásahy potom musely být větší. Ani jedna z těchto variant se u fuzzy regulátoru nestala. Kromě

toho bylo dosaženo 10 % úspory energie. I přes to však v soupravě zůstává pro případ nouze či poruchy strojvedoucí.

1.3.2 Jeřáb

Další významné uplatnění fuzzy regulace našla například u jeřábu, konkrétně u předcházení zhroupení nákladu [4]. Jelikož je nutné hlídat minimálně dvě vstupní proměnné (cílová poloha a úhel nosného lana), systém je nelineární, váha zátěže je neznámá a do systému může zasahovat porucha (vítr), návrh PID regulátoru je problematický, s čímž souvisí i cena regulátoru. Aplikace fuzzy regulátoru pro firmu Hochtief zvýšila rychlost transportu o 20 % a přispěla také k bezpečnosti, jelikož operátor jeřábu musel často chodit společně s nákladem a kontrolovat ho, často tedy klopýtal o překážky na zemi.

1.3.3 Systém ABS

Systém ABS zabraňuje zablokování kol auta při brždění ovládním tlaku brzdových destiček a tím zabraňuje smyku a zkracuje brzdovou dráhu. Systém ABS na základě informace z rychlosti otáčení jednotlivých kol vyhodnocuje, zda došlo ke smyku. I zde hraje roli spousta proměnných jako povrch vozovky a její sklon, stav pneumatik, jejich huštění a tak dále. Regulovaná soustava je tedy značně neurčitá. Klasické systémy se snaží udržovat poměr rychlosti otáčení kol a celkové rychlosti vozidla v jistém poměru, který je ovšem kompromisem pro různé situace. Systém fuzzy se však snaží přizpůsobit situaci a zefektivnit tím brždění.[9]

U některých systémů, například právě u ABS, není veřejně sdělováno, že byl systém fuzzy regulace použit, kvůli obavě společnosti, že by jeho název (mlhavé, vágní) mohl mít nepříznivý marketingový vliv.

1.3.4 Další použití

Lze najít spoustu dalších nejen průmyslových aplikací. Řízení natáčení vrtulí větrných elektráren pro maximální efektivnost, automatické převodovky, řízení teploty ve vstříkovacích lisech, omezovače rychlosti kamionů, inteligentní pračky a mnoho dalších (viz [9]).

Fuzzy logiku je možné použít nejen pro regulaci. Fuzzy logika měla úspěch například při řízení dopravy [9], zpracování obrazu (detekce hran, rozpoznávání objektu, ...) [10], [11]. Fuzzy logiku používá i software MASSIVE (Multiple Agent

Simulation System in Virtual Environment), který slouží pro 3D animaci davů a díky fuzzy logice se každý charakter chová do jisté míry individuálně [12].

1. 4. Fuzzy SW a HW prostředky

Fuzzy řízení je poměrně jednoduché implementovat ve většině programovacích jazycích. V této části bude představeno několik prostředků pro návrh fuzzy řízení.

FuzzyControl++[16] je balík nástrojů společnosti Siemens pro PCS 7 (ProcessControlSystem). Ten je určen především jako řídicí stanice pro řízení rozsáhlých systémů, obsluhující velké množství podružených podprocesů, jako je například elektrárna. Je zde i možnost propojit fuzzy logiku s neuronovými sítěmi pro optimalizaci procesu.

Společnost OMRON nabízí modul FZ001 do modulárního systému SYSMAC C200H [15]. Tento fuzzy modul může mít až 8 vstupů a 4 výstupy, přičemž každý z nich může obsahovat až 8 funkcí příslušnosti s 12 bitovým rozlišením. Počet inferenčních pravidel je omezen na 128. Čas jednoho strojového cyklu je do 6 ms. Jednotka se programuje přes PC, připojeném přes RS-232.

INFORM GmbH je jeden z předních softwarových vývojářů na fuzzy logiku. Jejich software FuzzyTech[13] umožňuje návrh fuzzy řízení a vygenerování odpovídajícího kódu v jazyce C, optimalizovaný assembly kód pro jednočipové počítače (mimo jiné například rodinu 8051 či PIC16CXX), návrh funkčních bloků pro Siemens S7 (PLC řady 300 a 400) nebo jako strukturovaný text pro CoDeSyS. Software spolupracuje se simulačními programy jako Matlab/Simulink.

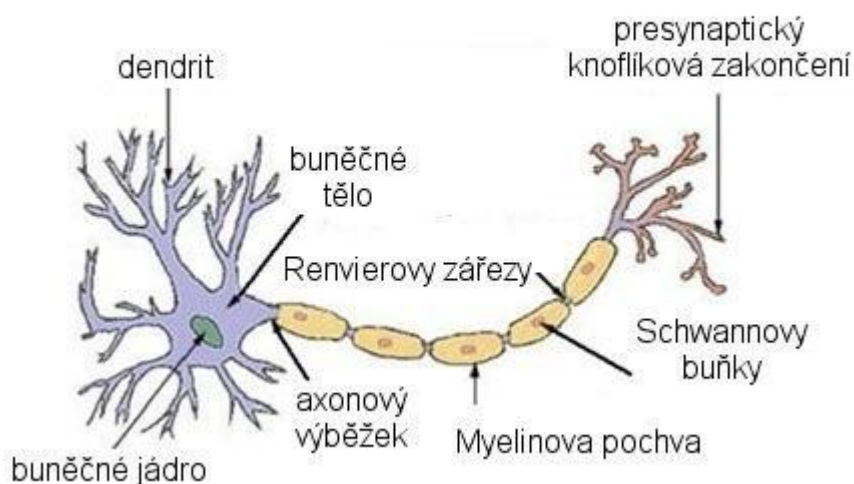
Aplikace fuzzylite je open-source knihovna naprogramovaná v jazyce C++ [14]. Umožňuje návrh a export kódu v jazycích C++, Java nebo jako soubor *.fis (používaný softwarem Matlab).

2. Neuronové sítě

Podobně jako u předchozí kapitoly, i zde budou nastíněny pouze základní znalosti a principy použité dále v práci. Při psaní kapitoly bylo čerpáno zejména z [17],[18] a [19], z anglicky psaných potom [20] a [21]. Tyto publikace lze doporučit i pro další rozšíření znalostí.

2.1. Biologický základ

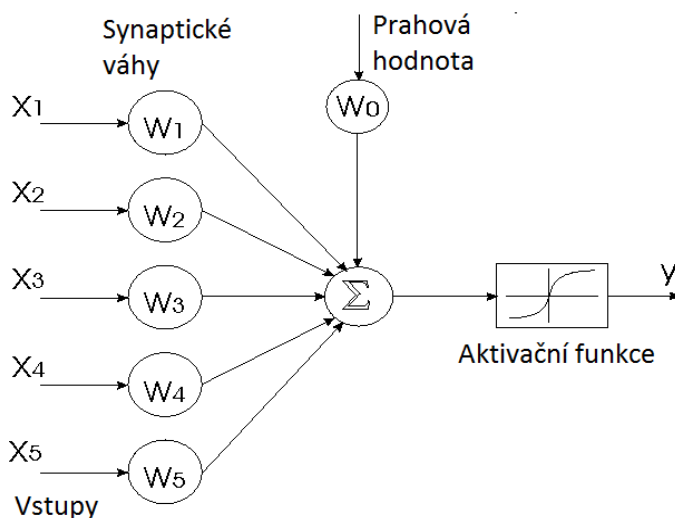
Jak bylo řečeno v úvodu, umělé neuronové sítě vychází z principu biologické neuronové sítě. Základním prvkem biologické neuronové sítě je jeden neuron.



2-1 Popis neuronu. Zdroj: [39]

Neurony se skládají z buněčného těla (sóma), z něhož vychází jediný výstup-axon. Ten je obalen membránou, díky které je možné šíření elektrické informace. Elektrický vzruch je přenášen na dendrity dalších neuronů synaptickými branami, které mohou být různě propustné a tím určují velikost podráždění „poslouchajícího“ neuronu. Právě díky změně synaptické propustnosti je zajištěna schopnost biologické neuronové sítě učit se. Překročí-li intenzita vzruchu na dendritech hraniční mez, dojde k vyslání elektrického impulzu axonem. Takových neuronů má kupříkladu lidský mozek přibližně 10^{11} [17], které jsou navzájem různě propojeny. Synaptické spojení se můžou s postupem času, jak vytvářet, tak i zanikat.

2. 2. Model neuronu

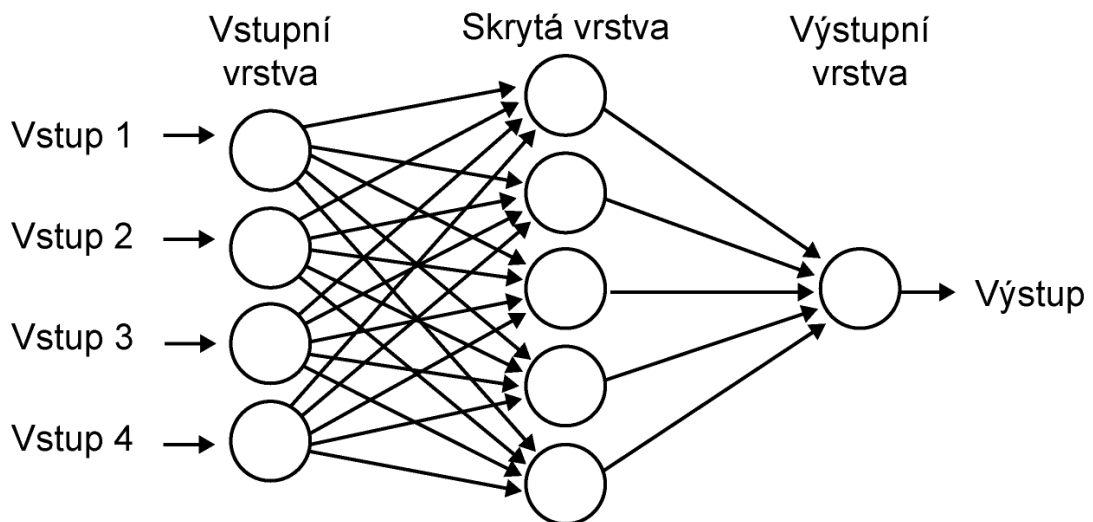


2-2 Počítačový model neuronu. Zdroj: [41]

Existuje více programových modelů neuronů (např. McCullochův model, viz [18]), v této práci se však pracuje výhradně s modelem tzv. perceptronu. Tento programový model neuronu se skládá ze vstupních signálů, váhových koeficientů (obdoba synaptických vah), prahovou hodnotou, sumátoru a aktivační funkce. Změna váhových koeficientů je podobně jako u biologického protějšku základem schopnosti učit se. Tento proces lze implementovat např. metodou zpětné propagace (BackPropagation), podrobněji popsanych např. v [18] a [19]. Jedná se o algoritmus, který upravuje váhové hodnoty, na základě rozdílu mezi spočítaným výstupem a požadovaným výstupem. Jedná se tedy o tzv. metodu učení s učitelem, kdy známe požadovaný výstup.

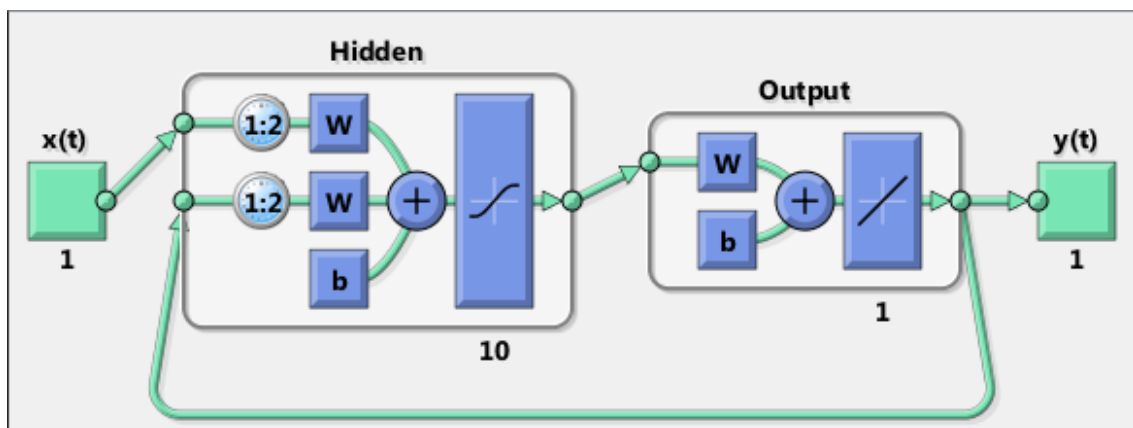
2. 3. Neuronové sítě

Základní strukturou propojení neuronů do neuronové sítě je dopředná vícevrstvá síť. Ta je rozdělená na tři části, takzvané vrstvy, mezi kterými je úplné propojení. To znamená, že každý výstup neuronu je vstupem do všech neuronů další vrstvy. Síť může být prakticky neomezeně velká, ve skryté vrstvě může být libovolný počet neuronů, existují i struktury s vícenásobnou skrytou vrstvou, ty jsou však již nad rámec této diplomové práce.



2-3 Programový model neuronové sítě. Každé kolečko na obrázku značí jeden perceptron, představený v předchozí podkapitole. Zdroj: [36]

Rozšířená struktura, použitá v této práci, je NARX (Nonlinear Auto Regressive network with eXogenous inputs) síť. První rozdíl mezi dopřednou a NARX sítí je, že do NARX sítě vstupuje nejen jeden vektor vstupních hodnot ale i několik vektorů z minulosti. Díky tomu síť dokáže lépe popsat dynamiku systému. Dalším rozdílem je, že výstup je připojen (také může být pomocí zpožďovačů přivedeny i vzorky z minulosti) zpět na vstup. Síť má tak informaci o svém předchozím stavu/stavech. Díky těmto úpravám oproti dopředné vícevrstvé síti je tato síť vhodná pro úlohy modelování dynamických systému, či pro predikci chování.



2-4 Náhled na NARX síť v softwaru MATLAB. $x(t)$ značí vstupní vektor, $y(t)$ výstupní vektor, blok 1:2 pak značí, že vstupují dva poslední časové vzorky vektoru. Je i patrná zpětná vazba z výstupu na vstup. Zdroj: [37]

2. 4. Aplikace neuronových sítí

2. 4. 1 Zpracování obrazu

Dle [22] je neuronové sítě možné použít prakticky na různorodé úlohy zpracování obrazu. V článku je zkoumáno, jaký typ sítě je vhodné použít na danou úlohu. Zkoumány byly úlohy jako například obrazová rekonstrukce, obrazová komprese, segmentace, rozpoznávání obrazu, porozumění obrazu. Ze článku vyplývá, že neuronové sítě mají ve zpracování obrazu jistě svou pozici, zejména pak v rozpoznávání, ale nevidí význam nasazovat neuronové sítě na všechny úlohy. Je zde zmíněno, že neuronové sítě by mohly najít uplatnění v úlohách, kde je vhodný adaptivní přístup, či pro rychlé úlohy, u kterých by se využilo paralelní počítání sítě.

V dalším článku [23] je využito neuronové sítě jako detektoru hran. Zde je zmíněno, že použitá technika měla dobré výsledky, zejména pak na obrázcích s vysokým rozlišením.

2. 4. 2 Chemický a farmaceutický výzkum

V publikaci [24] je několik příkladů použití neuronových sítí v chemickém, biochemickém, potravinářském a farmaceutickém výzkumu. Je zde například zmíněno použití samo organizační mapy, využívající neuronových sítí, pro proces virtuálního screeningu, tedy přístupu, který využívá metody molekulárního modelování a slouží k zefektivnění vývoje nových ligandů, nejčastěji léčiv [25]. Dále je zde uvedeno několik použití neuronové sítě jako prediktoru, např. pro predikci konstanty troposférické degradace organických sloučenin v ozónu, kde neuronová síť dosahovala lepších výsledků než algoritmus MLR (Multiple Linear Regression), dále pak predikci retenčního času peptidu kapalinové chromatografie, kde neuronová síť předpověděla v polovině testovací sady výsledky s chybou méně než 3 % a více než 95 % z testovací sady s chybou méně než 10 %. Podle dalšího výzkumu byla neuronová síť použita k predikci aktivity antioxidantů u různých druhů čaje, kde autor získal 99,9 % korelaci predikovaných a experimentálních dat.

2. 4. 3 ALVINN, Autonomně řízené vozidlo

Článek [26] již z roku 1989 a [27] pojednává o použití neuronových sítí k řízení autonomního vozidla. Vstupem do sítě byla kamera s rozlišením 30×32 bodů, ze které se vzaly hodnoty modré barvy (kvůli kontrastu silnice s okolím). Dalším vstupem byla matice 8×32 hodnot z laserového měřiče vzdálenosti. Skrytá vrstva neuronové sítě se

skládala z 29 neuronů. Výstupem bylo 45 hodnot, které udávaly míru zatáčení. Síť byla trénována tak, aby byl aktivní pouze jeden výstupní neuron pro daný úhel otočení volantu. Protože zde vystupovalo mnoho proměnných, vstupní údaje byly převedeny na generátor simulovaného obrazu cesty, který měl téměř 200 parametrů, např. pozice kamery, šířka silnice či velikost a směr stínu. Vozidlo bylo schopné učit se „za běhu“, tedy se nejdříve učilo od řidiče, pak řidič přepnul na autonomní řízení. K „naučení“ stačilo přibližně 5 minut. Díky učení s řidičem bylo vozidlo schopné naučit vypořádat se spoustou různých situací. Vozidlo bylo schopné zpracovat až 15 snímků za vteřinu, což mu dovolovalo jízdu až 55 mil v hodině (tedy necelých 90 kilometrů v hodině).

2. 4. 4 Další použití

Kromě výše zmíněných mají neuronové sítě spoustu dalších praktických využití. Například v práci [28] vypočítává se 70 % přesností pravděpodobnost výskytu minerálu v půdě. V práci [29] je neuronová síť použita k odhadu ekonomických dat. Článek [30] pojednává o použití neuronových sítí k rozpoznávání obličeje s úspěšností přes 90 %. A nakonec v práci [41] se neuronová síť zabývá předpovědí emisí CO₂.

2. 5. Identifikace dynamických systémů

V této kapitole bude provedeno několik simulací dynamických systémů, na kterých bude ověřena schopnost neuronových sítí naučit se jejich chování. Dále zde bude diskutován vliv parametrů neuronové sítě. K simulaci byl využit software MATLAB s toolboxem NeuralNetwork. Vybrané typy soustav byly zvoleny, jelikož mají přesah do praktické části.

2. 5. 1 Postup modelování systému pomocí neuronové sítě

K natrénování neuronové sítě je potřeba navrhnout identifikační signál. Měl by dostatečně postihnout dynamiku v celé pracovní oblasti. Obecně platí, že čím více dat se neuronové síti předloží k učení, tím bude výsledek lepší. Dlouhým identifikačním signálem se však prodlouží výpočetní čas k učení neuronové sítě, zejména potom u neuronových sítí s vysokým počtem zpožděných vektorů vstupů. Je nutné volit kompromis mezi výpočetním časem a výkonem sítě. Je tedy vhodné vystihnout chování systému v co nejkratším čase.

V této práci se nejlépe osvědčil signál vytvořený sumou dvou sinusových (či v kombinaci s chirp signálem) funkcí a bílého šumu. První, pomalejší sinus zajistí, aby se signál pohyboval v celé pracovní oblasti, tedy v čase identifikace proběhne jen

několik period přes celou pracovní oblast. Další, rychlejší sinus společně s namodulovaným bílým šumem má za úkol vystihnout dynamiku systému. Bílý šum obsahuje všechny frekvence, má tedy pozitivní vliv na identifikaci systému (nejen u neuronových sítí) a rychlý sinus zajistí, že se do každého bodu v pracovním rozsahu dostane několikrát s různou strmostí. Používaný identifikační signál lze tedy popsat rovnicí:

$$x[i]_{ident} = G_1 \times \sin\left(2\pi f_a \frac{i}{f_S} + \varphi_a\right) + G_2 \times \sin\left(2\pi f_b \frac{i}{f_S} + \varphi_b\right) + G_3 \times rand + K \quad (1)$$

Kde G_j = zesílení dané složky

K = stejnosměrná složka

F_j = frekvence j -tého sinusu

i = index vzorku

φ_j = fázové zpoždění j -tého sinusu

f_s = vzorkovací frekvence

Je zde několik možností trénování sítě v softwaru MATLAB. Lze využít toolboxů vyvolaných příkazy „nnstart“ či „nntool“, jelikož však pro potřeby práce bylo nutné cvičit několik neuronových sítí s různými parametry, byl na učení napsán skript. Jednu z modifikací skriptu lze nalézt v přílohách (Příloha G).

Nejdříve je nutné převést data do struktury využívající Neural Network toolbox softwaru MATLAB. Na to slouží příkaz „tonndata“. Dále je nutné zvolit parametry neuronové sítě. Ve všech případech byla použita síť typu NARX, to znamená, že kromě standartních vzorků vstupu je přivedeno i několik zpožděných vzorků (viz 2. 3). V této práci je zkoumán vliv těchto parametrů neuronové sítě:

- Počet neuronů ve skryté vrstvě
- Počet zpoždění vstupů
- Počet epoch učení
- Vliv náhodné inicializace váhových koeficientů
- Vliv velikosti dat k učení

Sítě jsou trénovány a zpracovávány ve třech vnořených for-cyklech. Další nastavení jsou parametry učení (trénovací metoda, rozdělení dat na trénovací a ověřovací, podmínky přerušení učení, ...), ty byly zanechány defaultní. Nyní lze spustit

učení příkazem „train“. Jelikož je výhodné učit s neuzavřenou smyčkou z výstupu na vstup NARX sítě, je nutné uzavřít smyčku až po učení, tedy příkazem „closeloop“.

Po natrénování sítě se ověřuje výsledek na kontrolním signálu. Kontrolní signál je nejčastěji série skokových funkcí s náhodnou amplitudou v pracovní oblasti. Výsledek je porovnán s očekávaným výstupem. Ověřen je v první řadě graficky, dále pak pomocí příznaku NRMSE (Normalized Root Mean Square Error), který se pohybuje v intervalu $(-\infty; 1]$. Čím lepší výsledek, tím více se blíží NRMSE číslu 1.

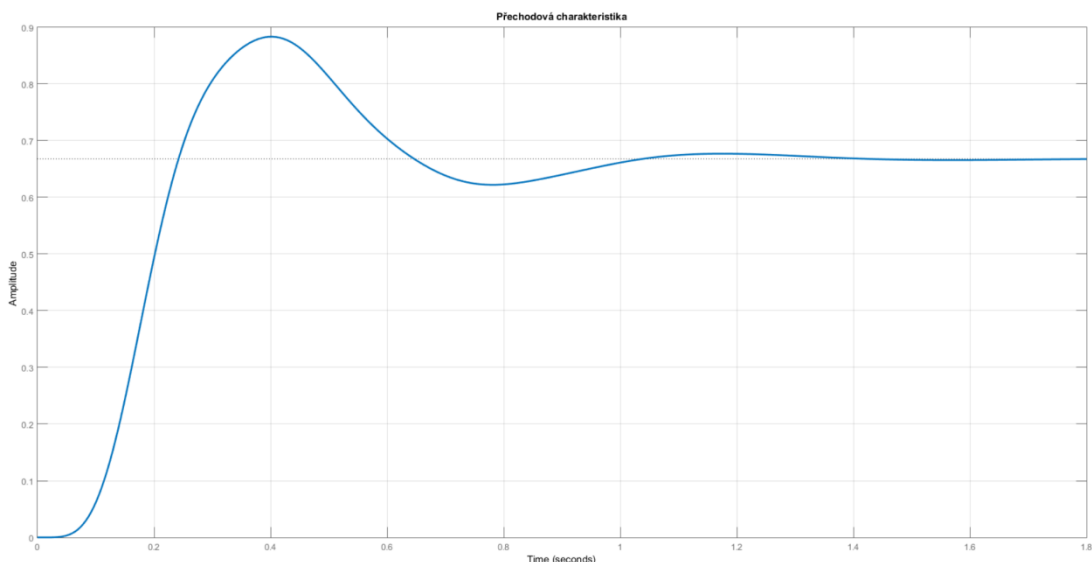
Co ovšem software MATLAB pravděpodobně neumožňuje je inicializace zpožděvacích členů. Neuronové sítě se tak zpočátku simulace chovají dosti nevyzpytatelně, což se negativně odráží mimo jiné na hodnocení NRMSE, které by při správné inicializaci bylo vyšší.

2. 5. 2 SISO systém

SISO model (Single Input Single Output) je zadán přenosovou funkcí:

$$F(s) = \frac{2 \times 10^6 \times (s + 8)}{(s + 16)(s + 20)(s^2 + 8s + 80)(s^2 + 12s + 936)} \quad (2)$$

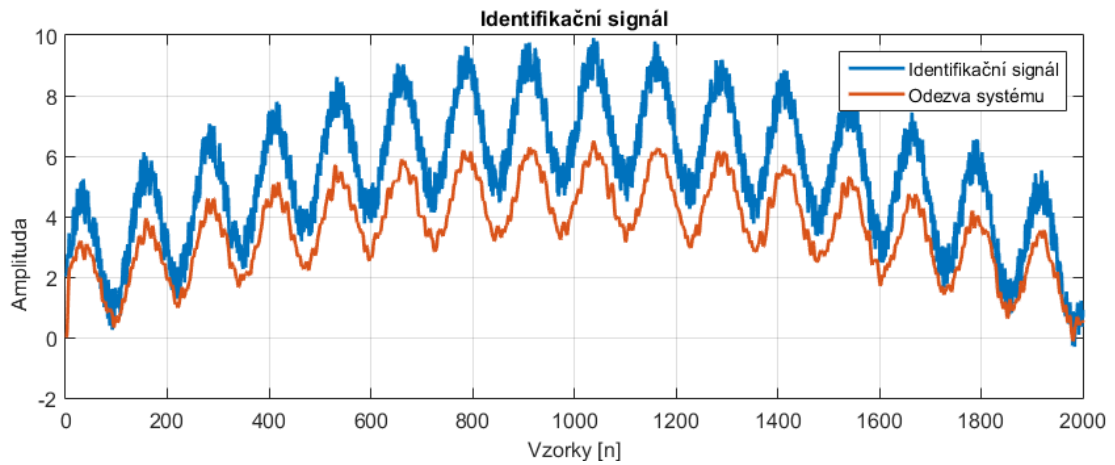
Jde tedy o dynamický systém 6. řádu s jednou nulou. Systém je stabilní.



2-5 Odezva SISO systému na jednotkový pulz

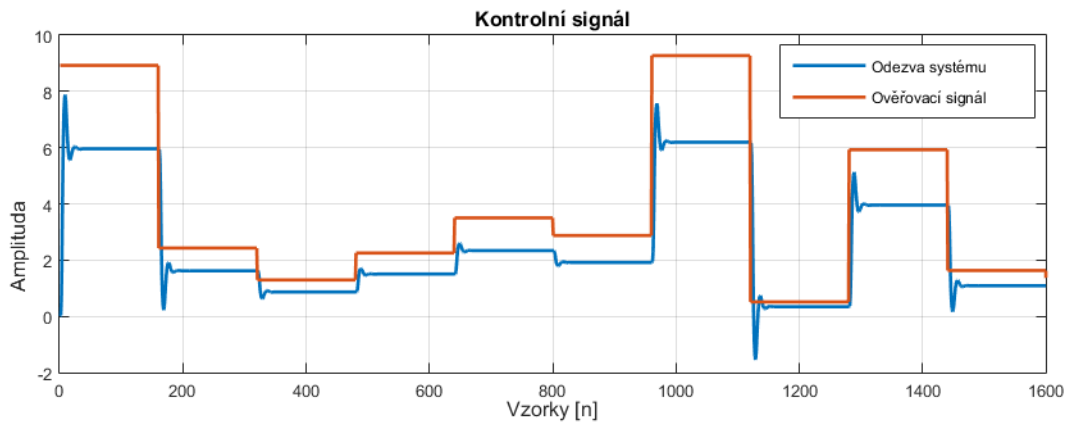
Postupovalo se dle kapitoly 2. 5. 1. Jako pracovní oblast byl zvolen interval 0-10. Identifikační signál je popsán rovnicí:

$$x_{ident} = 5 \sin\left(\frac{i}{f_s} \pi\right) + 2 \sin\left(\frac{i}{f_s}\right) + 1,5 * rand + 1,5 \quad (3)$$



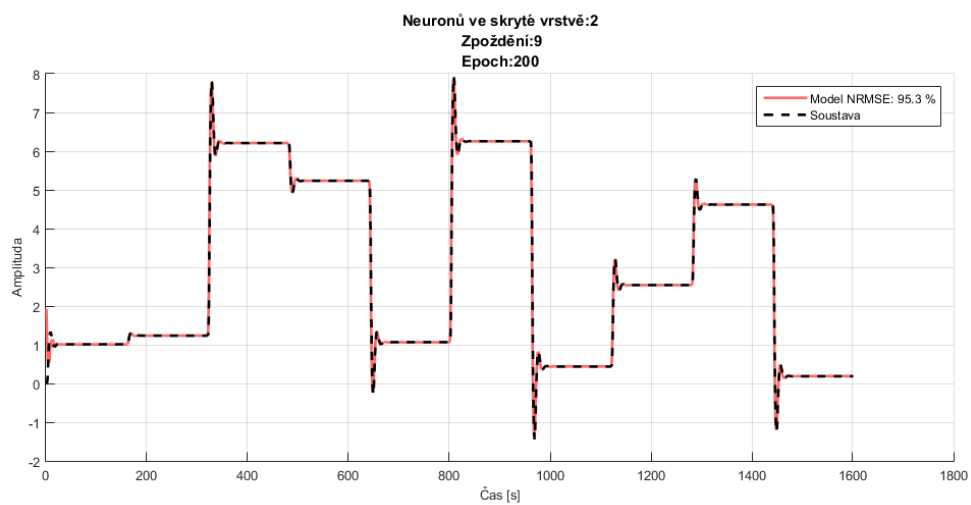
2-6 Identifikační signál a odezva SISO systému

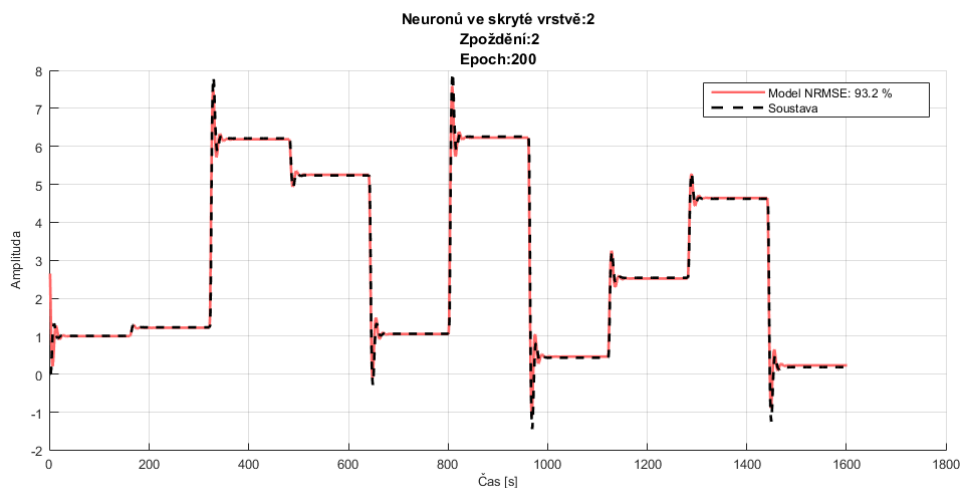
Dále byl vytvořen ověřovací signál a odezva modelovaného dynamického systému na tento signál.



2-7 Kontrolní signál a odezva SISO systému

Zde je několik výsledků neuronových sítí s různými parametry.

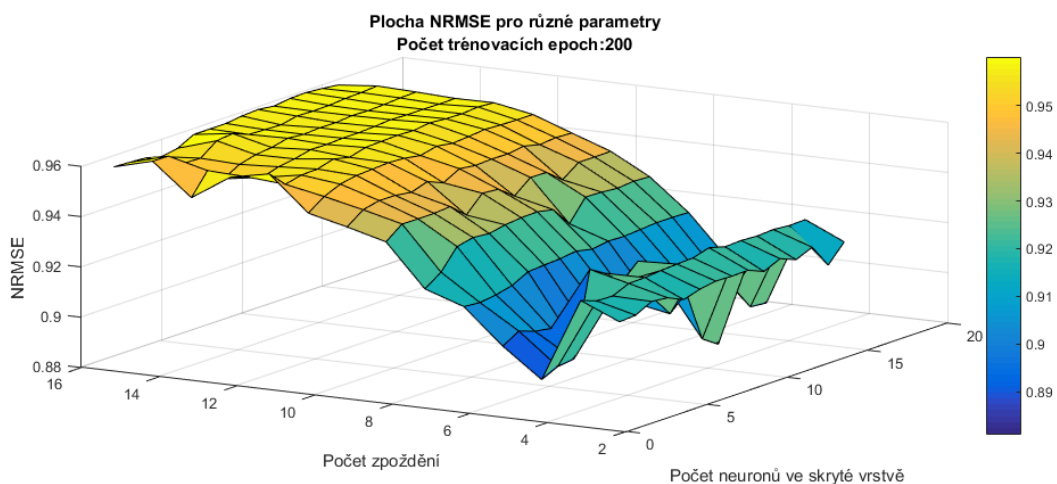




2-8 Porovnání dynamické soustavy a modelu

Z výsledků je patrné, že pro daný problém stačí i opravdu jednoduchá neuronová síť, i se sítí s dvěma neurony ve skryté vrstvě bylo dosaženo 96 % shody dle NRMSE.

Dále byla vytvořena matice 15×15 neuronových sítí ke zjištění vlivu parametrů. Konkrétně parametrů počet neuronů ve skryté vrstvě a počet zpoždění vstupu. Síť byla trénována jak po 200 učicích epoch, tak po 400 učicích epoch, ale jelikož byl rozdíl zanedbatelný, jsou níže prezentovány pouze výsledky pro 200 učicích epoch.



2-9 Přesnost neuronové sítě v závislosti na parametrech

Tabulka 2 NRMSE neuronových sítí u SISO systému pro 200 učicích epoch

NRMSE	Neuronů ve skryté vrstvě															
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Zpoždění	2	0,94	0,80	0,93	0,93	0,87	0,94	0,84	0,85	0,91	0,81	0,94	0,81	0,90	0,85	0,92
	3	0,93	0,94	0,93	0,93	0,93	0,93	0,93	0,94	0,94	0,94	0,93	0,92	0,94	0,93	0,94
	4	0,95	0,94	0,95	0,95	0,95	0,95	0,95	0,95	0,94	0,95	0,95	0,94	0,95	0,95	0,95
	5	0,93	0,93	0,93	0,91	0,89	0,90	0,91	0,92	0,92	0,94	0,93	0,93	0,92	0,93	0,93
	6	0,93	0,94	0,94	0,94	0,93	0,93	0,93	0,94	0,93	0,93	0,94	0,94	0,94	0,94	0,94
	7	0,94	0,94	0,94	0,94	0,94	0,94	0,94	0,94	0,94	0,94	0,95	0,94	0,95	0,95	0,94
	8	0,94	0,95	0,95	0,95	0,95	0,95	0,95	0,95	0,95	0,95	0,95	0,95	0,95	0,95	0,95
	9	0,96	0,95	0,95	0,95	0,95	0,96	0,95	0,96	0,96	0,95	0,95	0,96	0,96	0,96	0,96
	10	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96
	11	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96
	12	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96
	13	0,95	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96
	14	0,96	0,96	0,95	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96
	15	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96
	16	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,96	0,97	0,96

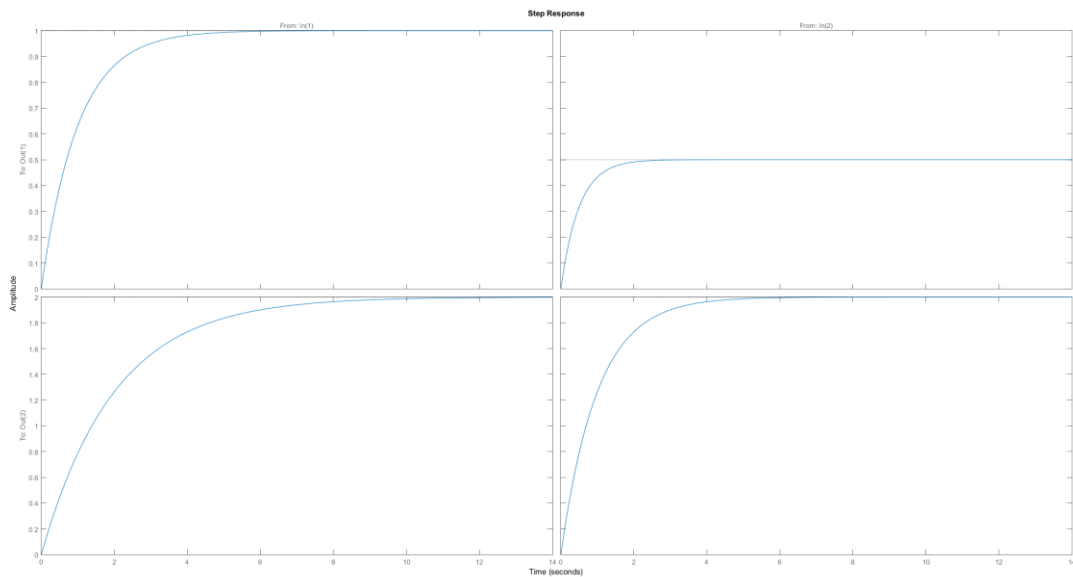
Z povrchu vytvořeného z přesností sítí lze usoudit, že významnější vliv pro tuto soustavu má počet zpoždění. Počet neuronů ve skryté vrstvě neovlivňuje výsledek tak významně, u sítí s větším počtem zpoždění již skoro vůbec. Dále si lze všimnout lokálního minima u sítí s pěti zpožděnými vzorky na vstupu. Lze pouze spekulovat, proč tento jev nastal, zda je to učicím algoritmem, dobou učení, či není tato struktura vhodná pro daný typ problému. Porovnáním hodnot z tabulky a grafu si lze všimnout, že byl záměrně vynechán první řádek (tedy s dvěma zpožďovacími členy). Jelikož se v tomto řádku objevilo několik hodnot vzdálených od většiny, grafické zobrazení tak mírně ztratilo na přehlednosti.

2.5.3 MIMO systém

MIMO (Multiple Input Multiple Output) systém je popsán přenosovou maticí:

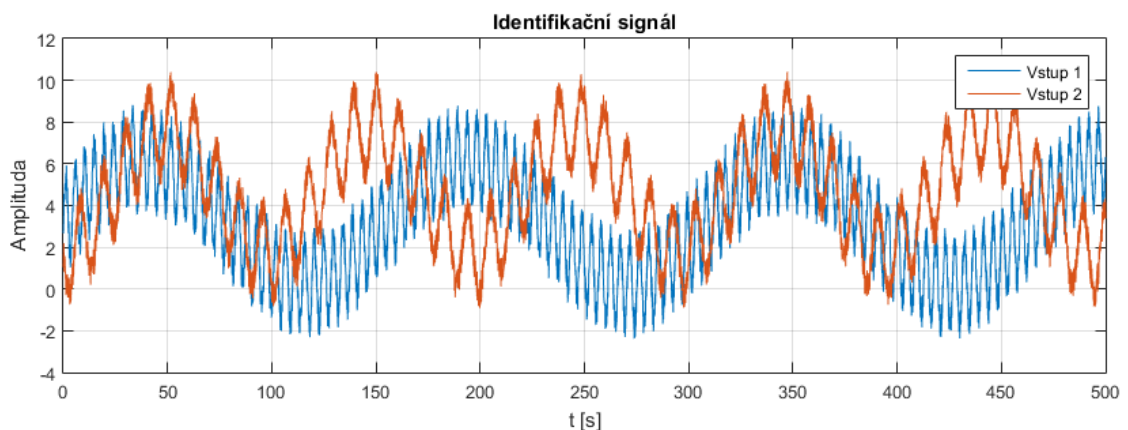
$$F(s) = \begin{bmatrix} \frac{1}{s+1} & \frac{1}{s+2} \\ 1 & 2 \\ \frac{1}{s+0,5} & \frac{1}{s+1} \end{bmatrix} \quad (4)$$

Jde tedy o systém, který má dva vstupy a dva výstupy. Jak přímé, tak křížové vazby jsou popsány přenosovou funkcí prvního řádu, což je patrné i z přechodové charakteristiky (obr. 2-10).

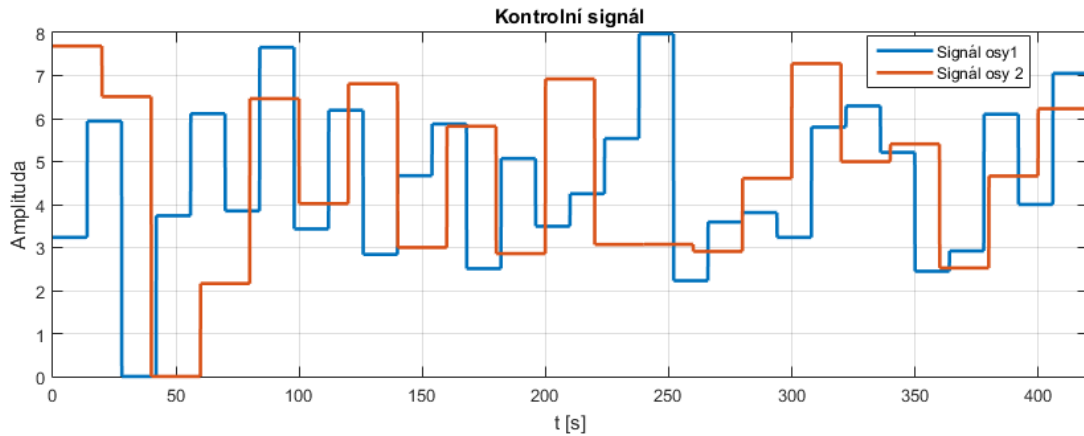


2-10 Odezva MIMO systému na jednotkový pulz

Jako pracovní rozsah byla zvolena oblast mezi 0-10. Další postup je podrobněji popsán v 2. 5. 1. Rozdílné je pouze to, že systém má dva vstupy, je nutné vytvořit identifikační signál jako vektor o dvou složkách. Pokud by byl zanechán původní identifikační signál a vstupoval by shodně na oba vstupy, nebyly by popsány křížové vazby. Je tedy nutné vytvořit signál takový, aby se každá ze složek vstupního signálu dostaly v jeden okamžik do různých oblastí pracovního rozsahu. Proto byly zvoleny různé úhlové rychlosti jednotlivých složek vstupního signálu. Identifikační signál by měl zasáhnout v celé pracovní oblasti, dovoluje-li to aplikace, je vhodné navrhnout identifikační signál mírně přes hranice pracovní oblasti. Jako kontrolní signál byla opět zvolena náhodná posloupnost skoků.

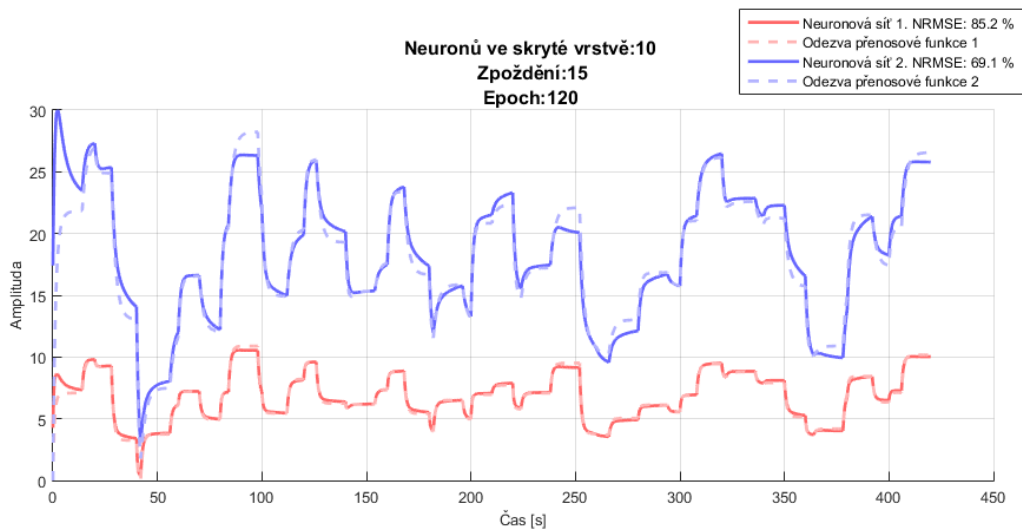


2-11 Identifikační signál pro MIMO soustavu



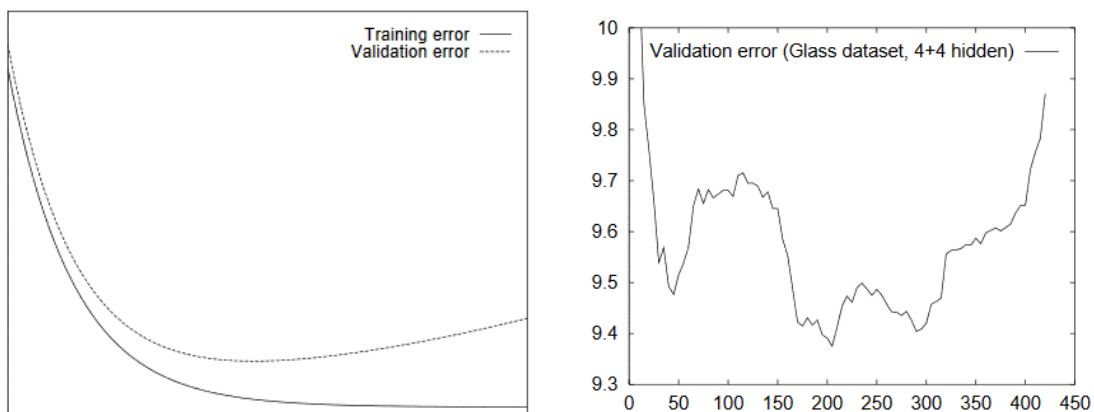
2-12 Kontrolní signál použitý na MIMO systém

Poté bylo trénováno několik neuronových sítí s různými parametry (počet neuronů ve skryté vrstvě, počet zpoždění na vstupu). Díky tomu bylo možné zjistit strukturu sítě, která dokáže dostatečně popsat systém.



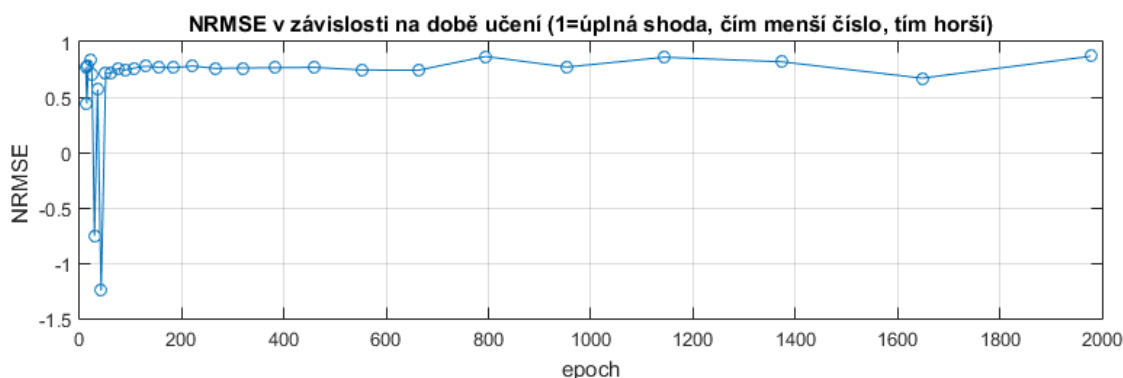
2-13 Porovnání MIMO systému a modelu

Dále byl na tomto systému sledován vliv doby učení neuronové sítě. Krátká doba učení by zřejmě neměla vést k uspokojivému výsledku, naopak dlouhá doba učení by měla způsobit tzv. přeučení, tedy že síť se naučí velice dobře identifikační signál, ale jeho schopnost reagovat na obecný signál se sníží. O přeučení se zabývá například článek [35], kde je možné najít idealizovaný a reálný průběh přeučení (viz obrázek 2-14 níže).



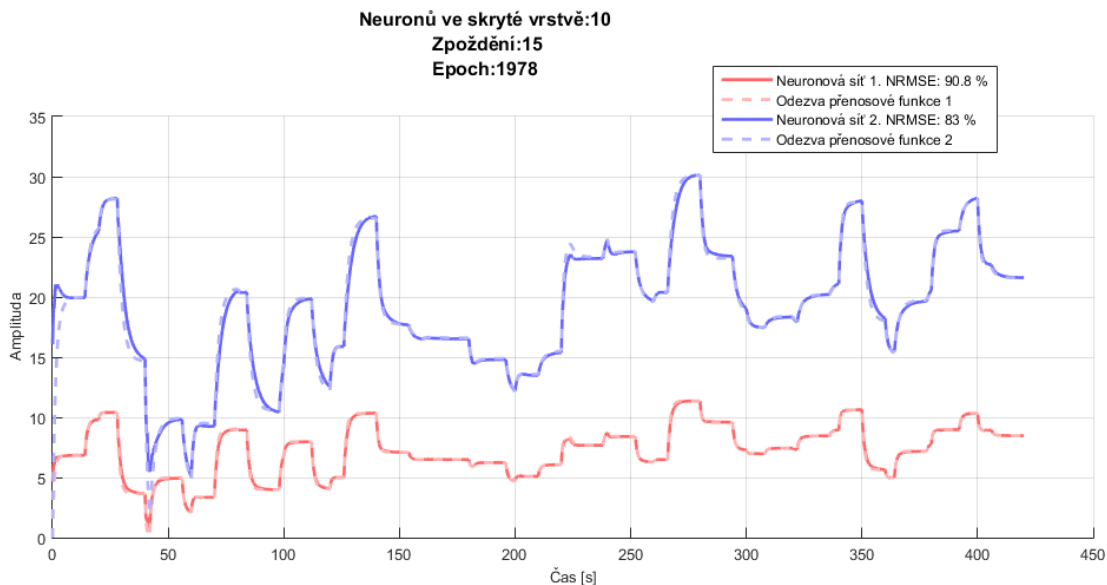
2-14 Teoretický a reálný průběh přeučení neuronové sítě. Zdroj [35].

Na prvním obrázku lze vidět, že zatímco chyba trénovací sady klesá, v jednom momentu chyba kontrolní sady začne růst. Nejnižší bod trénovací sady je tedy nejlepším kandidátem, kdy učení zastavit. Vpravo je potom vidět příklad reálného průběhu přeučení, kde se tento ideální bod zastavení učení nachází v lokálních minimech a lze se tak dostat do dalších minim s lepší i horší přesností. Bod ideálního přerušení učení tedy není tak jednoznačný.



2-15 Závislost přesnosti sítě na počtu epoch

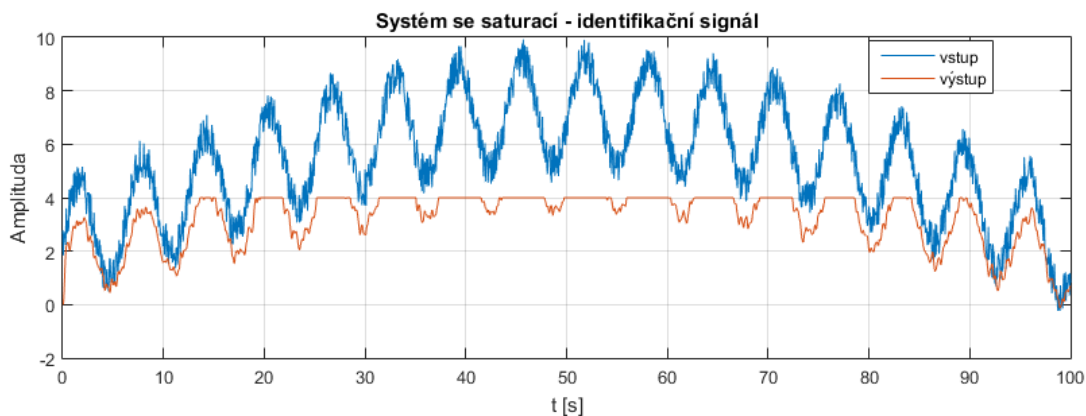
V provedeném rozsahu nedošlo k přeučení sítě. Kromě lokálních minim v počátku učení byl výsledek všech sítí přes 70 % shody NRMSE. Naopak poslední síť (nejvíce naučená) vykazovala největší přesnost. Přesto si lze v grafu všimnout malých výkyvů, které charakterem odpovídají reálnému průběhu přeučení na obrázku 2-14. Nelze tedy tvrdit, že s rostoucím počtem epoch se výsledky pouze zlepšují, lze však říci, že se výsledek s největším počtem epoch nalézal právě v lokálním minimu. Jelikož má systém dva výstupy, byl v tomto případě počítán kvantifikátor přesnosti NRMSE pro každý výstup zvlášť a celkové NRMSE jako jejich aritmetický průměr.



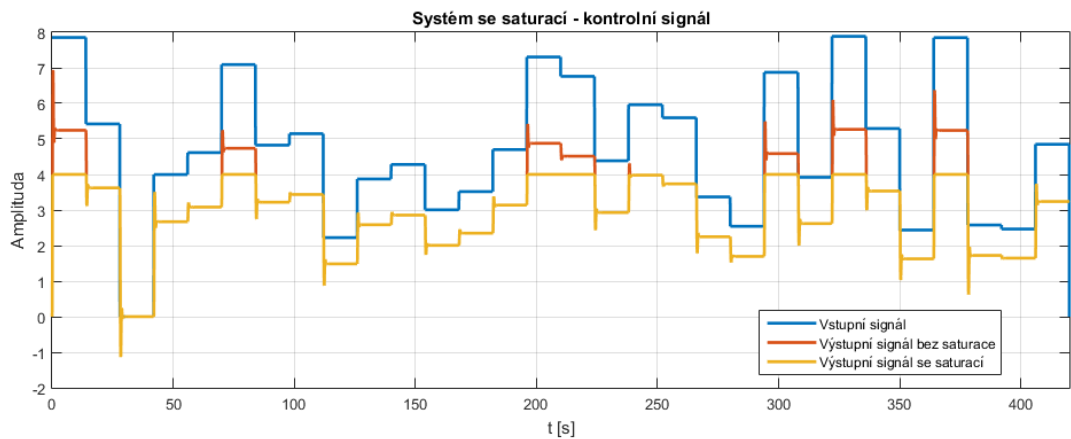
2-16 Síť s největší přesností (dle NRMSE)

2. 5. 4 Systém se saturací

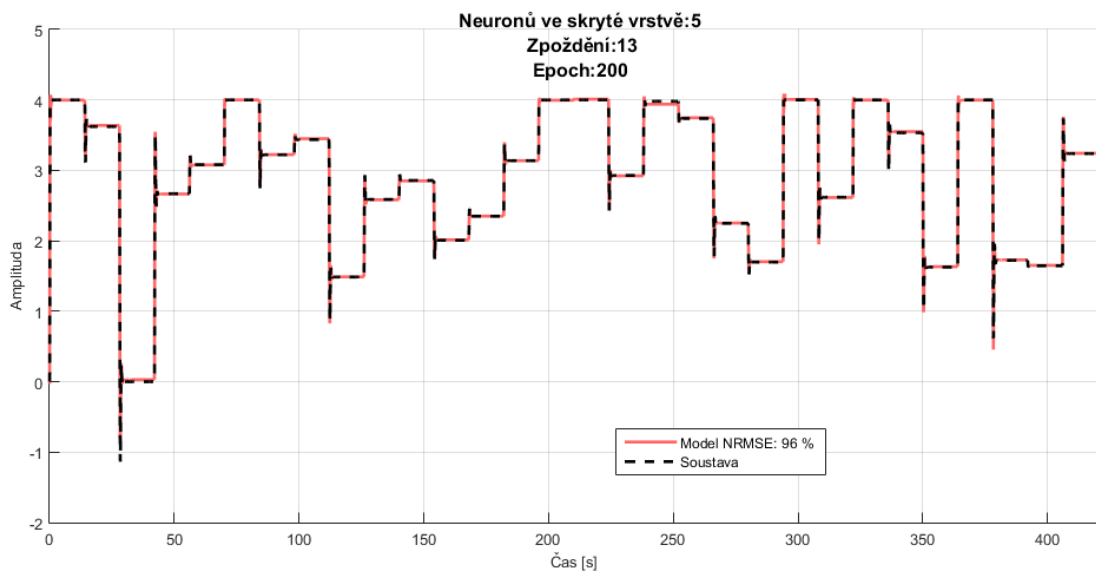
Další dynamický systém k ověření funkce neuronových sítí je systém se saturací, tedy takový systém, který dosáhne maxima (či minima), přes které se dále nedostane. Lze si to představit jako objem vody v otevřené nádrži, která začne přetékat, či doraz u kinetického systému. Byl využit systém z kapitoly 2. 5. 2 se saturací na hodnotě čtyři. Na obrázku 2-17 je zobrazen identifikační signál, na dalším obrázku (2-18) je porovnání odezvy systému se saturací a systému bez saturace na kontrolní signál.



2-17 Identifikační signál systému se saturací



2-18 Kontrolní signál, odezva systému na kontrolní signál. Červenou barvou je naznačena odezva systému bez saturace. Lze si všimnout saturace na hodnotě 4, kterou systém bez saturace (červená barva) překročí.



2-19 Porovnání modelu a odezvy soustavy

Z výsledků na obrázku 2-19 lze pozorovat, že odezva na kontrolní signálu modelu oproti simulovanému systému je prakticky totožná, tedy že i systém omezený touto nelinearitou na výstupu lze pomocí neuronových sítí s velkou přesností modelovat. Tento systém byl použitý k zjištění vlivu inicializace hodnot učení.

Učení neuronové sítě začíná s náhodnými velikostmi váhových koeficientů, což může způsobit, že učení může za stejných vstupních podmínek dopadnout různě. Tento výsledek je ovlivněn především učícím algoritmem (rychlost a způsob konvergence), přesto lze obecně říci, že opakované učení nepovede na totožný výsledek, pouze může konvergovat do stejného lokálního minima. Tento jev lze pozorovat na tabulce níže (Tabulka 3). Na tabulce je vidět, že i přes shodné parametry učení i stejnou strukturu

sítě, bylo nalezeno v hodnotě přibližně 0,66 jedno lokální minimum. Čtyři sítě se inicializovaly takovými hodnotami, které v daném počtu učicích epoch nedokázaly najít lokální minimum.

Tabulka 3 NRMSE z opakovaného učení

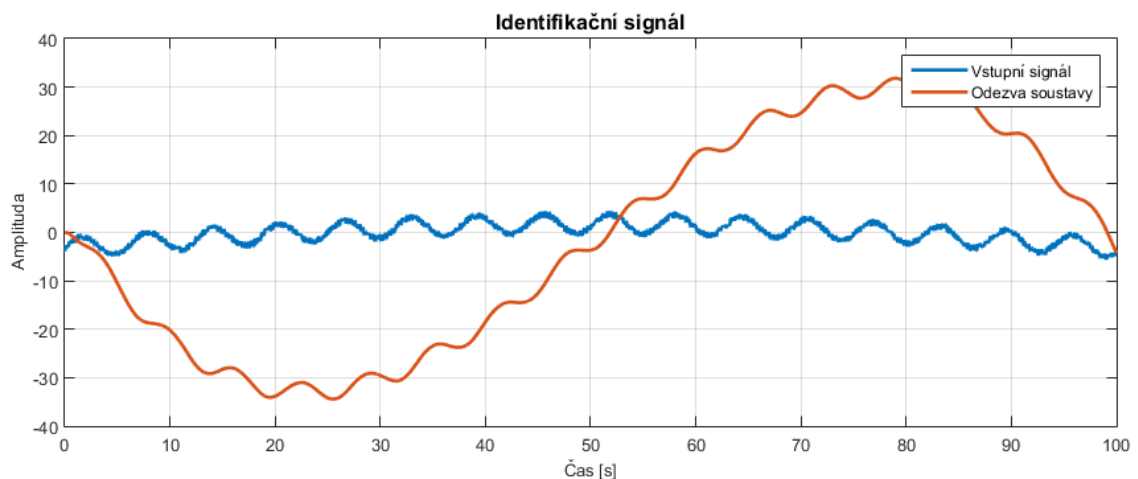
Číslo opakování	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NRMSE	0,68	0,67	0,68	-7,52	-2,09	0,68	0,67	-7,17	0,66	0,63	0,64	-7,77	0,68	0,68	0,68

2. 5. 5 Astatický systém

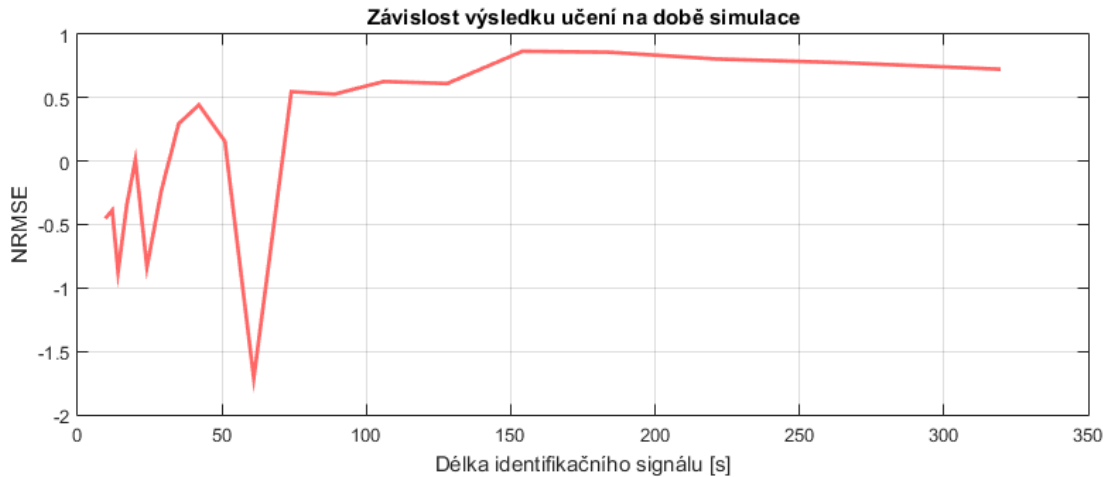
Astatický systém, tedy systém s alespoň jedním pólem ležícím v počátku souřadnicového systému komplexní roviny, je popsán rovnicí:

$$F(s) = \frac{1}{s(1 + 0.51s + 0.0676s^2)} \quad (5)$$

Identifikační signál byl zanechán podobný jako u předchozích příkladů, pouze byl posunut i do záporných čísel. Podobně i kontrolní signál zůstal zachován s tím rozdílem, že generovaná náhodná čísla nejsou pouze kladná. Rozšířený pracovní rozsah o záporná čísla byl zvolen kvůli přehlednosti grafů. Díky integračnímu charakteru by odezva neustále stoupala a nízké hodnoty by měly nízkou vypovídající hodnotu.



2-20 Identifikační signál astatického systému



2-23 Závislost učení na délce identifikačního signálu

Tabulka 4 Výsledek NRMSE v závislosti na délce simulace

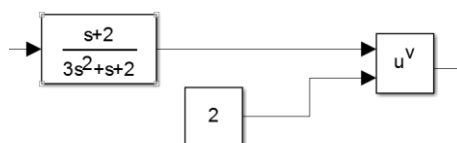
Čas simulace	10	12	14	17	20	24	29	35	42	51
NRMSE	-0,44	-0,38	-0,87	-0,34	0,01	-0,84	-0,23	0,29	0,44	0,15
Čas simulace	61	74	89	106	128	154	184	221	266	319
NRMSE	-1,71	0,55	0,53	0,63	0,61	0,87	0,86	0,81	0,78	0,73

Z výsledku je patrné, že velikost identifikačních dat ovlivňuje celkový výsledek. I s přihlednutím faktu představeném v předchozí kapitole o náhodné inicializaci, je zde vidět, že sítě s dostatkem informací dávají lepší výsledky. Lze si však všimnout klesající tendence v závěru grafu, která bude nejspíše souviset s počtem učících epoch.

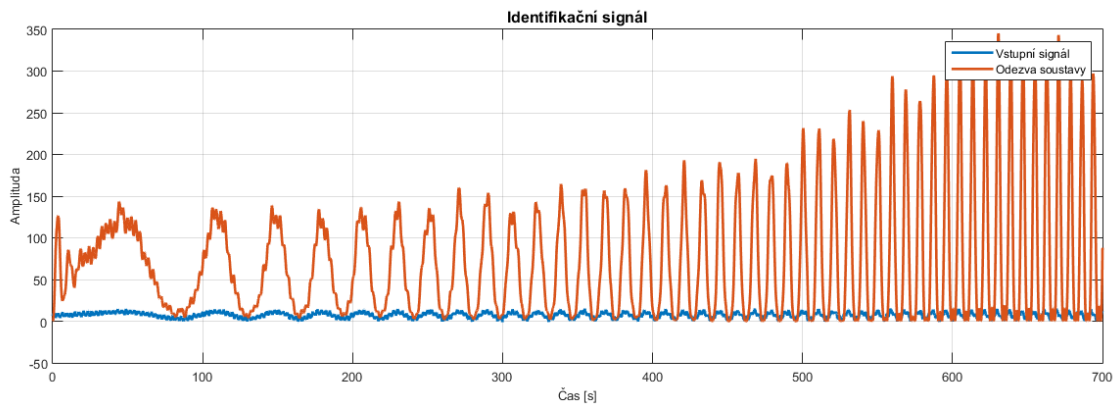
2. 5. 6 Nelineární systém

Nelineární je takový dynamický systém, u kterého neplatí princip superpozice. Bude-li tedy přiveden na vstup systému signál A a poté signál B, bude u lineárního systému platit, že výstupní signál bude shodný jako by byl přiveden na vstup systému součet signálu A a B. To u nelineárního systému nemusí platit.

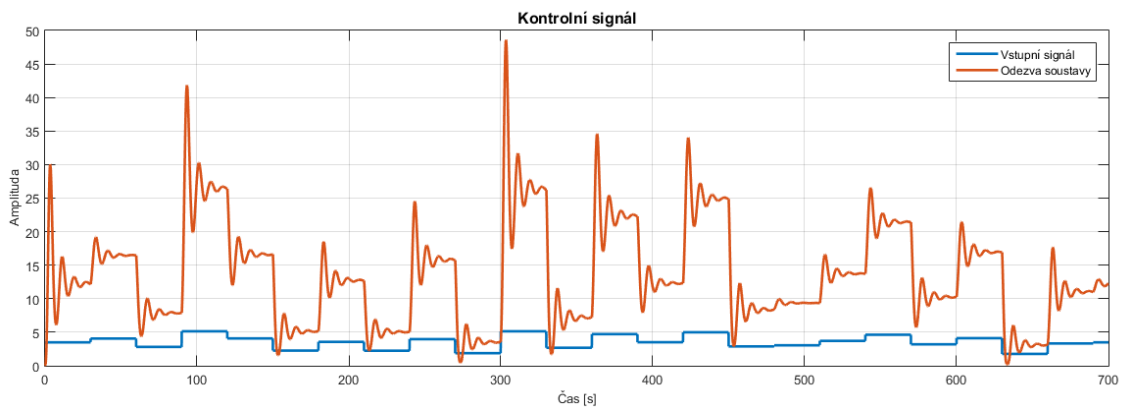
V práci byl zvolen LTI systém, jehož výstup byl umocněn na druhou. Podobně jako v předchozích kapitolách, i zde se vycházelo z postupu popsaném v kapitole 2. 5. 1.



2-24 Použitý nelineární systém

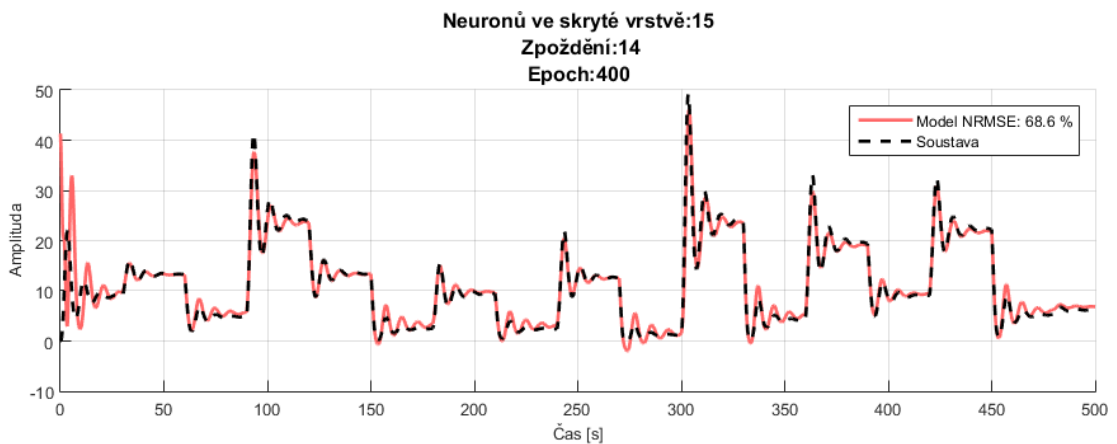


2-25 Identifikační signál a odezva nelineárního systému



2-26 Kontrolní signál a odezva nelineárního systému

I v tomto případě dosáhla neuronová síť velmi dobrých výsledků. Odezva se ustaluje na velmi podobné hodnotě, odpovídá frekvence kmitání, pouze velikost kmitů se v jistých místech grafu mírně liší. Opět byla vytvořena matice sítí a kromě několika výjimek se většina sítí k žádané dynamice ani nepřiblížila. Proto lze tvrdit, že oproti ostatním typům systémů byla úloha nelineárního systému nejnáročnější.



2-27 Porovnání modelu a odezvy soustavy

2.5.7 Shrnutí

V předchozích kapitolách byla simulačně provedena identifikace na několika typech dynamických soustav. Ačkoliv jde o čistě teoretické modely, byly zvoleny, jelikož mají přesah do praktické části. Tato část však měla především otestovat, zda jsou neuronové sítě vhodný prostředek pro úlohu modelování dynamických systémů, a získat alespoň základní představu o volbě parametrů neuronové sítě, tvaru identifikačního signálu a o možných komplikacích, zapříčiněných ať už neuronovou sítí obecně, či implementací v použitém softwaru.

Neuronové sítě podávaly ve všech simulacích velice dobré výsledky. Grafická podoba se simulovaným systémem je zřejmá. První úskalí však vzniká počátkem simulace, kdy nedochází k inicializaci hodnot zpožděvacích členů na vstupu. Neuronové sítě se tak v počátku nachází v nedefinovaném stavu, což způsobí v lepším případě nevyzpytatelné chování v prvních několika okamžicích, v horším případě se odezva rozkmitá a nedojde k ustalování v pracovní oblasti. Následkem toho dochází k diskreditaci výsledků snížením klasifikátoru NRMSE nebo k úplnému kolapsu sítě.

Vzhledem k provedenému testu, který dokázal, že síť s identickou strukturou, parametry a identifikačním signálem podává různé výsledky, je nutné brát některé výsledky s rezervou, pro opravdové sledování vlivu parametrů by bylo potřeba opakovat několikrát učení se stejnými podmínkami. Dále by bylo nutné sledovat i vzájemné vazby mezi jednotlivými parametry. S ohledem na počet parametrů (a ne všechny byly v práci zmíněné, lze pokračovat výběrem učící metody, poměr rozdělení identifikačního signálu na trénovací a ověřovací, vícenásobné skryté vrstvy, tvar aktivační funkce, vliv vzorkovací frekvence...) a výpočetní náročnost nebyla tato problematika rozebrána do podrobností.

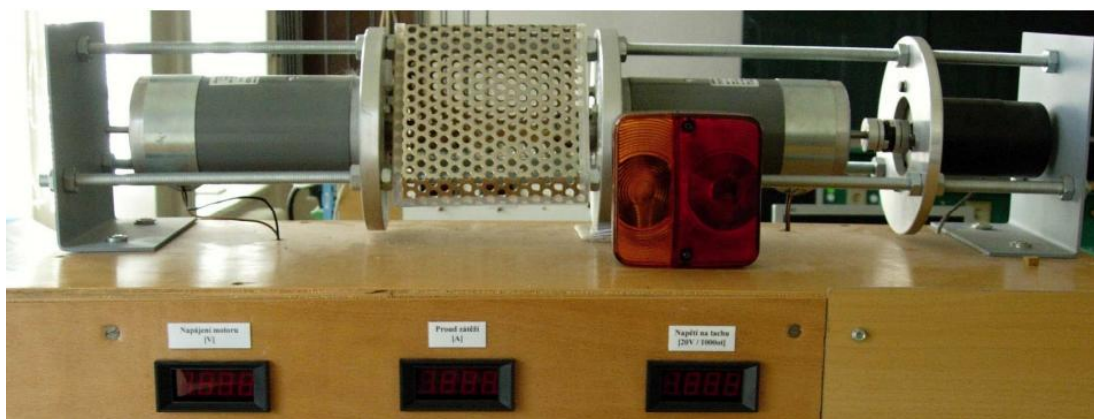
Povědomí o chování neuronových sítí bylo však dostatečně vytvořeno. Například bylo zjištěno, jaký je nejvhodnější tvar identifikačního signálu a že je potřeba identifikační signál navrhnout v celé pracovní oblasti, nejlépe i mírně přes její hranice. Naopak není vhodné identifikovat stavy, které prakticky nenastanou. Dále je pro popis dynamické soustavy nutný dostatečný počet zpožděvacích členů na vstupu NARX sítě, nebo že v softwaru MATLAB nedochází k přeučení, patrně díky parametrům učení, které tomu dokáží předejít.

3. Popis fyzických modelů

V simulačních modelech se prakticky ověřila funkčnost neuronových sítí na určitých typech dynamických soustav. V této kapitole budou představeny modely, na kterých budou tyto znalosti aplikovány.

3.1. DC motor

Soustava je tvořena dvěma stejnosměrnými elektromotory. První má roli zdroje energie soustavy a přes pružnou spojku je připojen druhý elektromotor, jehož hřídel je pevně spojena s tachodynamem. Vstupem do soustavy je tedy budicí napětí prvního motoru, výstupem napětí na tachodynamu, které je úměrné otáčkám druhého motoru. Oba motory jsou shodné, s maximálním napětím 24 V. Vstupní napětí je upraveno přes buzení motoru (charakteristika systému druhého řádu), v programu je vstupní rozsah 0-5 V. Výstupní napětí je upraveno přes dělič napětí, programový rozsah je 0-10 V. Další informace viz [32].



3-1 Model stejnosměrného motoru ve školní laboratoři

3.2. Fén

Hlavními prvky této úlohy je ventilátor, topná spirála a teplotní čidlo Pt100. Topná spirála je zapojena přes výkonový člen s pulsně šířkovou modulací do měřicí karty počítače, kde je možné výkon ovládat v rozsahu 0-10 V. Ventilátor se zapíná mechanickým tlačítkem, otáčky ventilátoru nelze řídit. Výstupem soustavy je teplota vzduchu, snímaná teplotním čidlem. Odpor čidla, závislý na teplotě, je převeden na proud 4-20 mA, poté dalším převodníkem na napětí 0-10 V, tedy již vstupem do měřicí počítačové karty. Pro podrobnější informace viz [33].



3-2 Laboratorní model fénu s teplotním čidlem

3. 3. Vrtulník CE 150

Vrtulník CE 150 od firmy Humusoft se skládá z těla, hlavního motoru, který tělo klopí nahoru a dolů, pomocného motoru, který tělo otáčí doleva a doprava. Dohromady má tedy model dva stupně volnosti, je však možné šroubem zaaretovat osu a libovolný stupeň volnosti tak odebrat. Momenty, které vyvíjejí stejnosměrné motory ovlivňují druhou osu řízení, kupříkladu otáčení hlavního motoru způsobuje rotaci do strany, vytváří se tak křížová vazba mezi vstupy a výstupy. Podrobnější popis, včetně fyzikálního modelu lze nalézt například v [34]. Úhel naklopení a natočení je snímán inkrementálním senzorem. Model je připojen k počítači laboratoře přes měřicí kartu.



3-3 Vrtulník CE 150. Zdroj:[38]

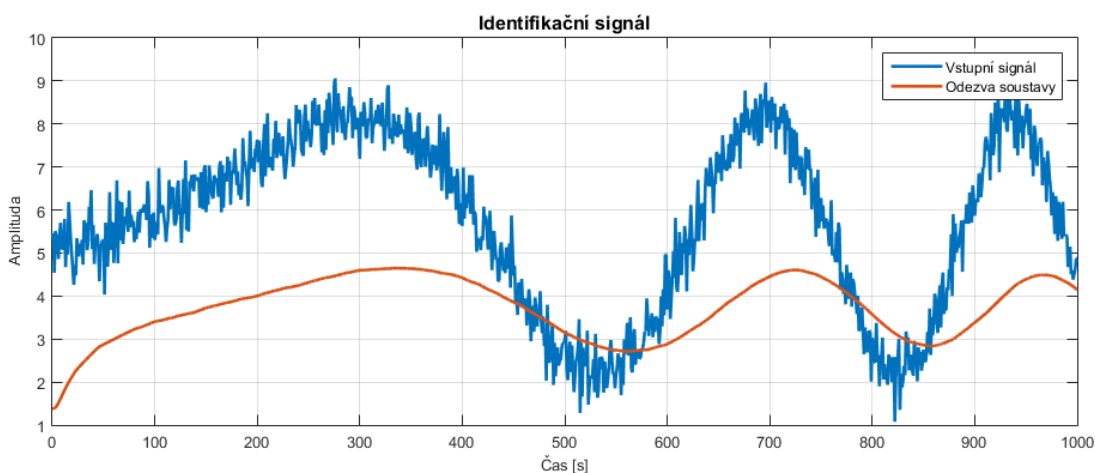
4. Praktická část

V této části se práce zaměří na praktické využití teoretický znalostí z předchozích kapitol. Znalosti fuzzy regulace a identifikace pomocí neuronových sítí z kapitol 1 a 2 budou aplikovány na laboratorní modely popsané v kapitole 3.

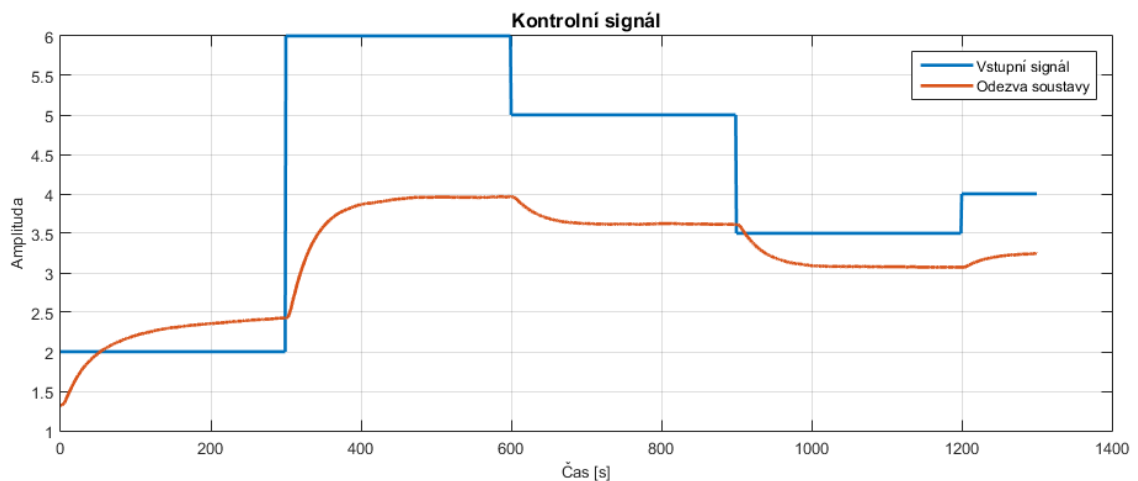
U identifikace se vycházelo z metody popsané v kapitole 2. 5. 1, případně její modifikaci. Jak bylo také ve zmíněné kapitole naznačeno, v softwaru MATLAB patrně není možnost inicializace zpožděvacích členů neuronové sítě, proto je jejich chování v počátku simulace nepředvídatelná, což se projevilo právě zejména v praktické části. Hodnocení NRMSE by za řádné inicializace bylo větší a chování sítě alespoň v počátku přesnější.

4. 1. Identifikace soustavy – fén

Pro identifikaci byl použit signál na obrázku 4-1, tedy chirp signál s namodulovaným bílým šumem. Jelikož se jedná o tepelnou soustavu s dlouhou časovou konstantou, byla zvolena vzorkovací frekvence s periodou jedné vteřiny. Kontrolní signál (obrázek 4-2) byl vytvořen ze série skokových změn o náhodné hodnotě v pracovním rozsahu.

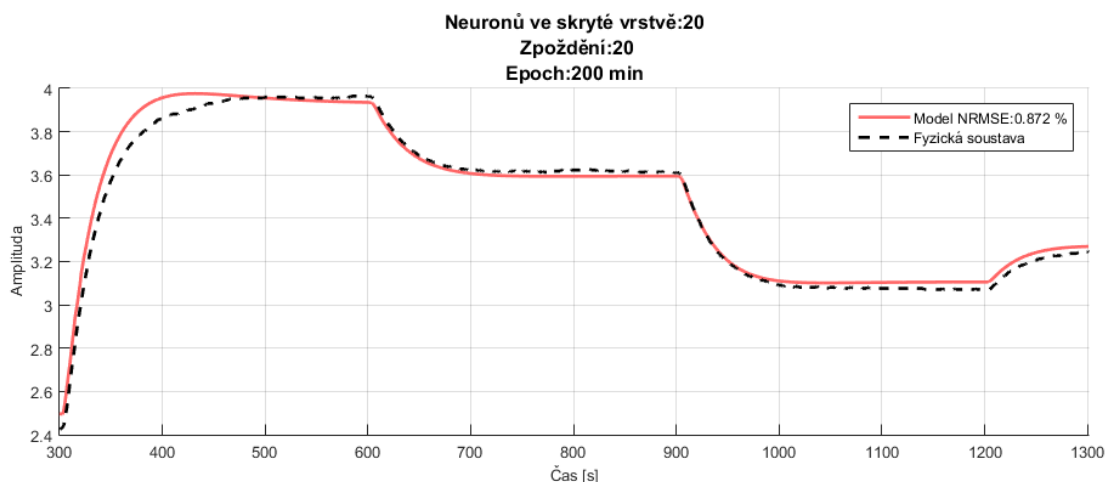


4-1 Identifikační signál a odezva systému pro laboratorní model fěnu



4-2 Kontrolní signál a odezva systému modelu fěnu

Díky zvolené vzorkovací frekvenci bylo trénování neuronových sítí velice rychlé (neuronová síť se učila pouze na 1000 vzorcích). U soustavy se jako menší problém projevila počáteční hodnota, jelikož se po každém měření nacházela v jiném počátečním stavu. Proto byl po každém měření ponechán čas na vychladnutí. Z výsledků je zřejmé, že úspěšnost neuronových sítí není vysoká (většina okolo 50 % shody dle NRMSE), což je dáno krátkou dobou učení (počet epoch). Při tak nízkém počtu epoch je výsledek velice závislý na inicializaci neuronové sítě a často zůstane neuronová síť v lokálním minimu. Nejlepší výsledek měla neuronová síť s 20 neurony ve skryté vrstvě a s 20 zpožďovacími členy na vstupu s výsledkem přes 87 %. Porovnání na obrázku 4-3 ukazuje, s přihlédnutím na fakt s počáteční hodnotou, velice dobrou shodu.

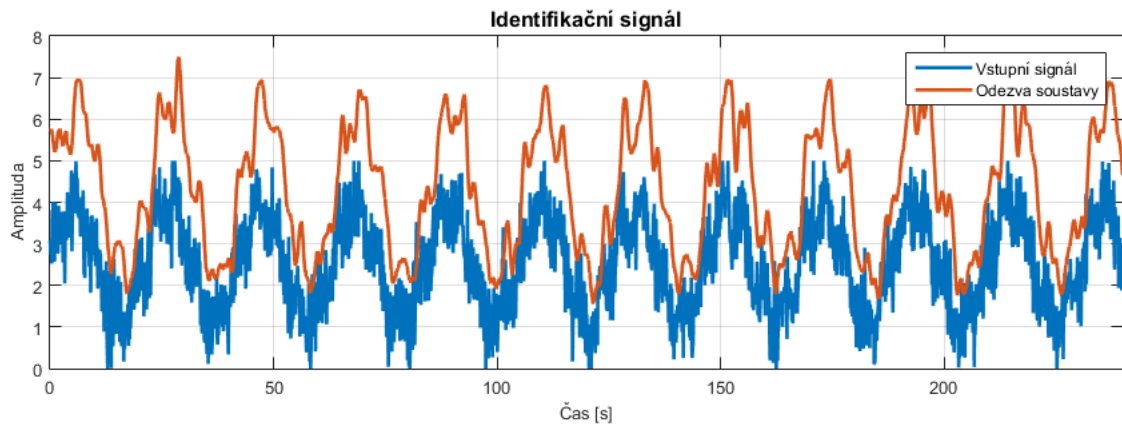


4-3 Porovnání fyzické soustavy motoru s modelem

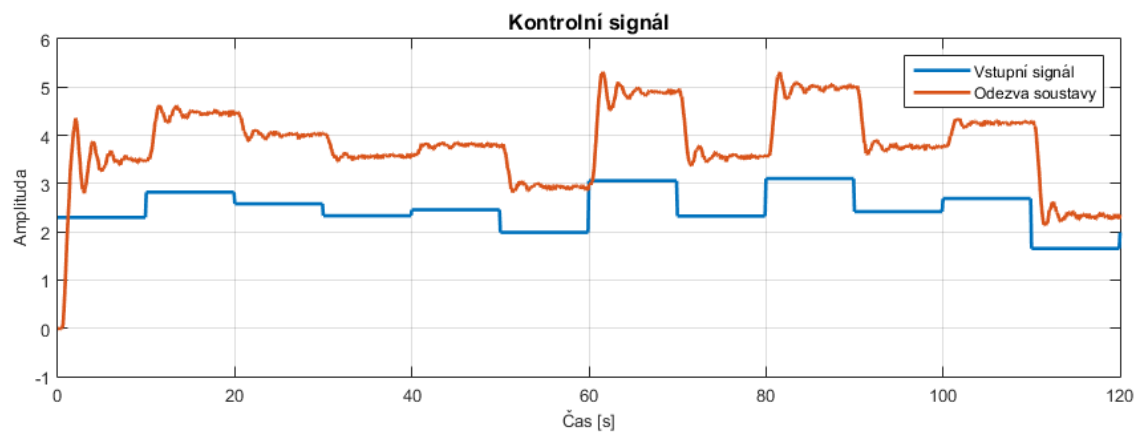
4. 2. Identifikace a řízení soustavy - motor

K identifikaci soustavy pomocí neuronových sítí byl použit postup popsany v kapitole 2. 5. 1. Jako identifikační signál byla zvolena suma dvou sinusových signálů

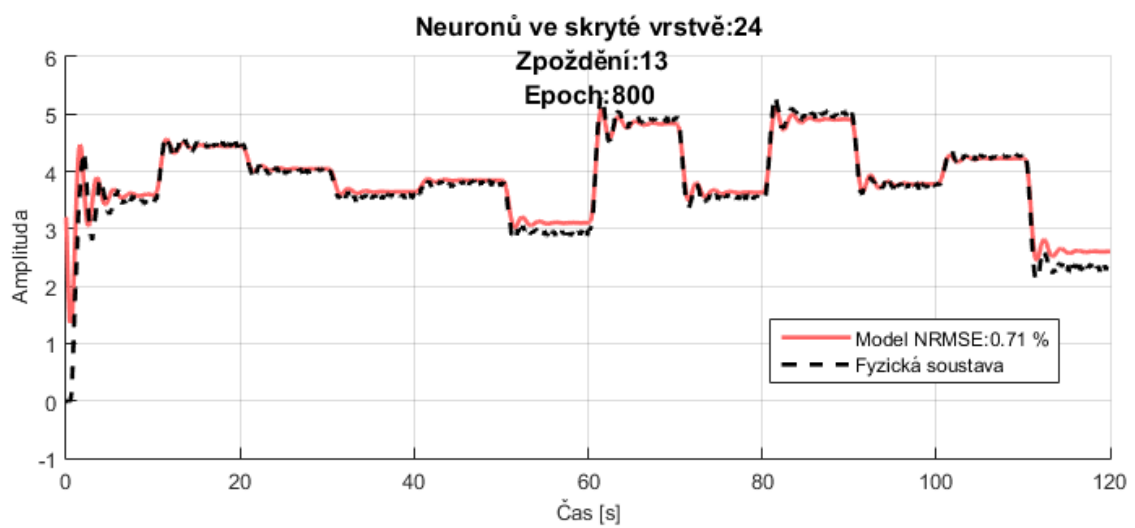
a bílého šumu (obr. 4-4), jako kontrolní signál série náhodných skoků v pracovní oblasti (obr. 4-5).



4-4 Identifikační signál a odezva systému pro model motoru a tachodynamama



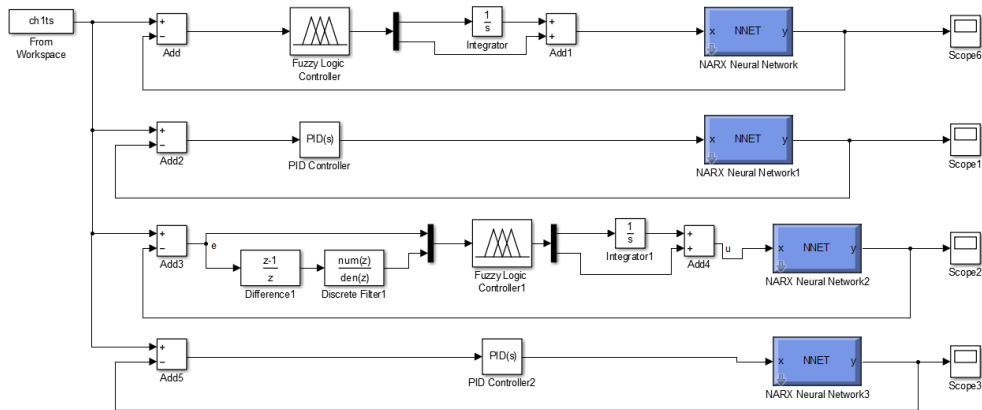
4-5 Kontrolní signál a odezva systému na model motoru a tachodynamama



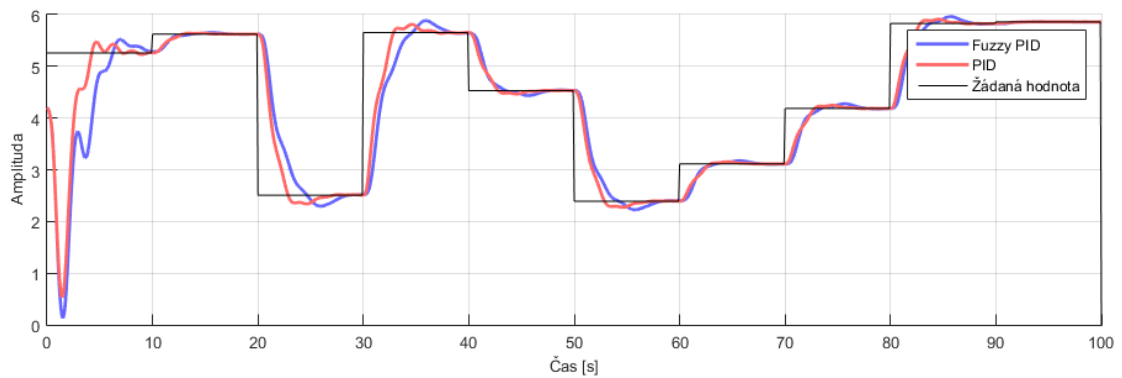
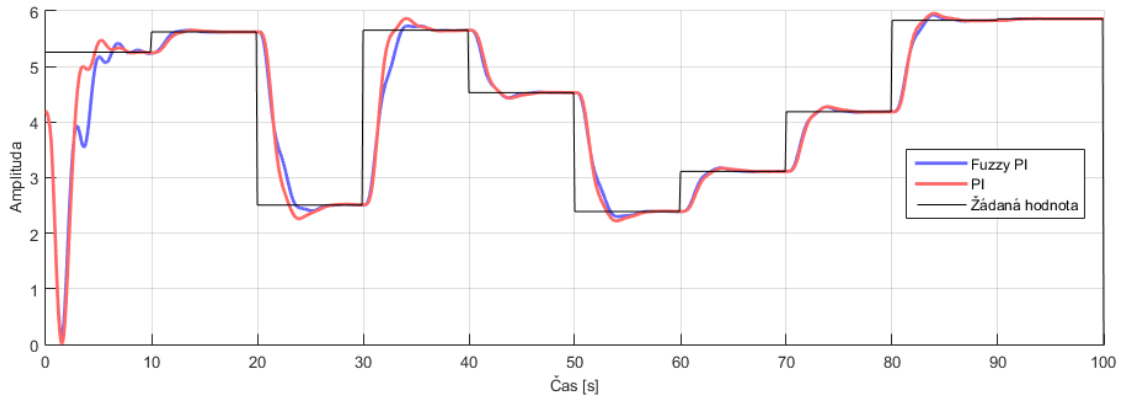
4-6 Výsledek učení neuronové sítě na úloze motoru

Z výsledku 4-6 je patrné, že neuronová síť identifikovala systém s poměrně vysokou přesností. Bylo dosaženo přesného popisu kapacitní vazby mezi motorem a tachodynamem a to jak frekvence, tak i amplitudy vznikajících kmitů. V některých místech pracovního rozsahu, zejména při nízkých hodnotách, však síť vykazuje chybu statickou, kde se neustaluje na přesné hodnotě.

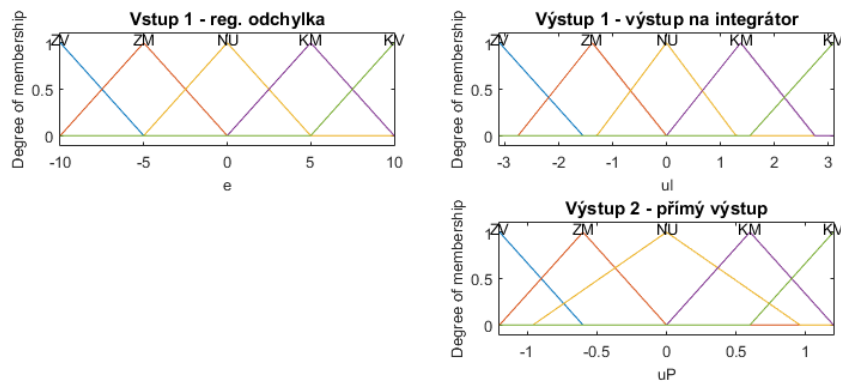
Na takto získaném modelu byla navržena regulace, a to jak konvenčním PI(D) regulátorem, tak i fuzzy přístupem. V nástroji Simulink programu MATLAB bylo vytvořeno schéma. Nejdříve byl navržen empirickou metodou PI regulátor, poté fuzzy regulátor (obr. 4-8). Autor se záměrně snažil věnovat oběma metodám podobné množství času, aby nedošlo k zvýhodnění žádné z metod. Tento experiment tedy měl zejména porovnat rychlost návrhu fuzzy regulátoru oproti PID regulátoru.



4-7 Schéma zapojení pro regulaci na modelu vytvořeném neuronovou sítí. Postupně odshora: fuzzy PI, PI, fuzzy PID, PID.



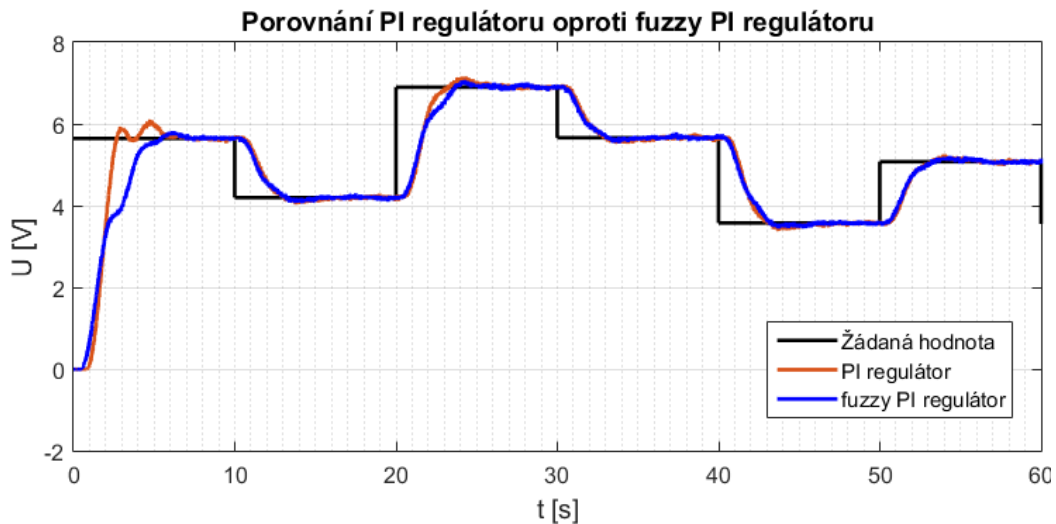
4-8 Návrh regulátoru na simulačním modelu. Model je tvořen neuronovou sítí. Na obrázku výše regulátor PI a jeho fuzzy obdoba. Na obrázku níže PID regulátor a fuzzy protějšek.



4-9 Funkce příslušnost fuzzy PI regulátoru

Na obrázku 4-9 jsou zobrazeny funkce příslušnosti PI regulátoru. Díky symetrii trojúhelníkových funkcí je zajištěno, že fuzzy regulátor přechází plynule mezi inferenčními pravidly, nevznikají tak žádné razantní změny akční veličiny. Naopak u P složky fuzzy regulátoru si lze všimnout rozšíření funkce příslušnosti „NU“. To zajistí opatrnější zásah regulátoru s klesající velikostí regulační odchylky. Byl použit

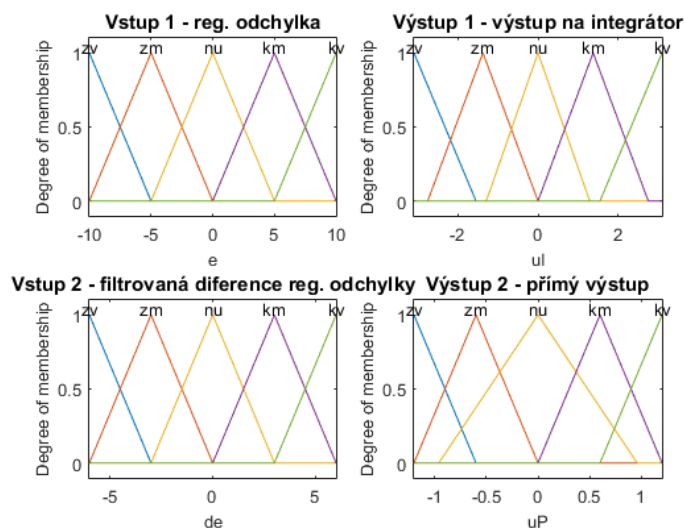
fuzzy regulátor s jednoduchou inferenční tabulkou s pěti pravidly. Takto navržené regulátory byly otestovány na fyzickém modelu.



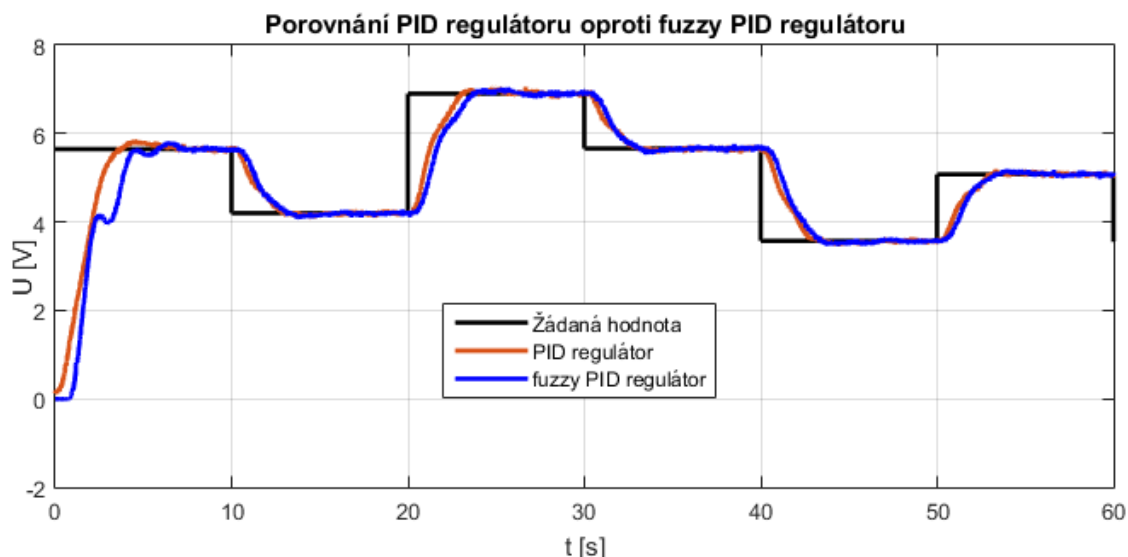
4-10 Porovnání regulace PI regulátoru a fuzzy PI regulátoru na fyzickém systému

Z výsledku je patrné, že na této dynamické soustavě je rozdíl minimální a postupnými úpravami by se dalo docílit pravděpodobně obdobného výsledku. V principu by mělo být možné nastavit fuzzy regulátor, aby reagoval stejně jako PI regulátor, jelikož lze fuzzy regulátor seřadit tak, aby obě složky reagovali na regulační odchylku téměř lineárně. Zavedením nelinearity by mělo být možné výsledek zlepšit, nebo minimálně lépe přizpůsobit požadavkům aplikace (rychlejší náběh/bez překmitu, ...).

Podobně se postupovalo u PID regulátoru. Obdobu derivační složky byla zavedena do fuzzy regulátoru vstupem filtrované diference, tj. rychlosti změny regulační odchylky. Byl proto empiricky navržen filtr dolní propust 4. řádu, s mezní frekvencí 15 Hz, který odfiltruje nespojitosti diskrétního vstupu. Funkce příslušnosti jsou zobrazeny na obrázku 4-11, regulátor byl vytvořen s 19 inferenčními pravidly.



4-11 Funkce příslušnosti fuzzy PID regulátoru



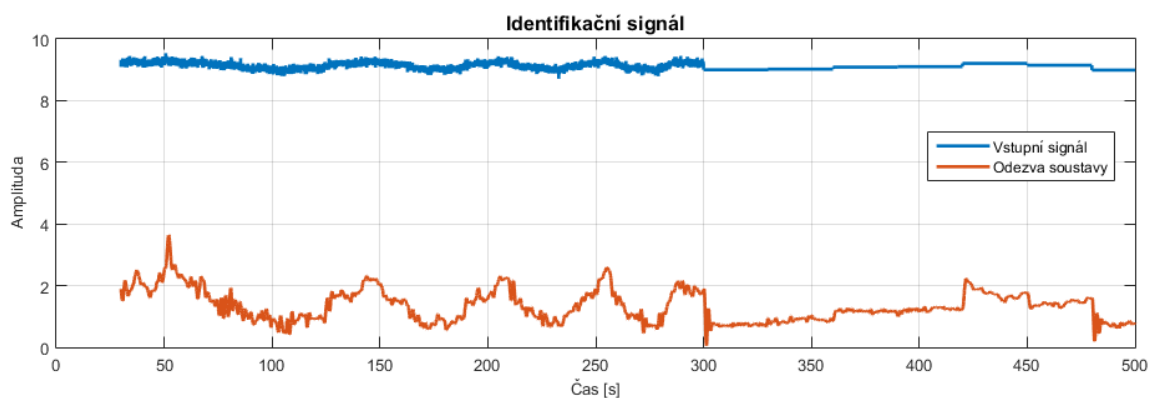
4-12 Porovnání regulace PID regulátoru a fuzzy PID regulátoru na fyzické soustavě

Doba nastavení fuzzy regulátoru se stejným výsledkem je přibližně obdobná jako u PID(PI) regulátoru. U fuzzy regulátoru je však více stupňů volnosti, je tedy více prostředků, jak celkový regulační pochod vylepšit. U PID je možné změnit pouze koeficienty, filtr u derivační složky a v jistých případech vzorkovací frekvenci. Pro další vylepšení je potřeba pokročilejší struktura (IMC, ...). U fuzzy regulátoru je možné měnit polohu, šířku, tvar, počet funkcí příslušnosti, dále je možné měnit inferenční pravidla a jejich váhu. Tím lze vytvořit prakticky libovolnou závislost výstupu jednotlivých složek na regulační odchylce. Fuzzy regulátor je tedy možné více přizpůsobit požadavkům aplikace. U jednoduchých soustav je tedy benefit fuzzy

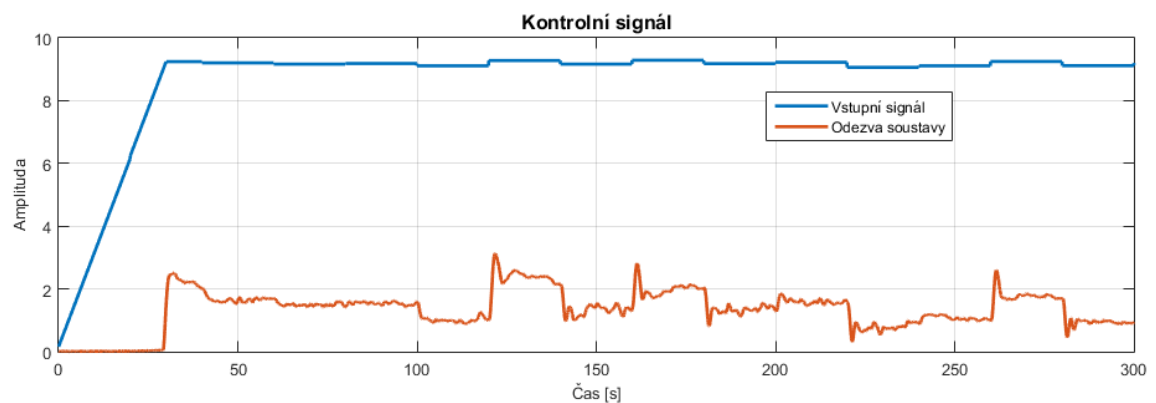
regulátoru menší, ale roste se složitostí soustavy, což bude demonstrováno na příkladu vrtulníku.

4. 1. Identifikace a řízení soustavy – vrtulník

Jako poslední model byl identifikován vrtulník. Model byl identifikován jako SISO, se zaaretovanou jednou osou, obecně jde o MIMO model. Vykazuje prvky nelinearity, astatismu, a díky dorazům lze mluvit také o saturaci, tedy všechny vlastnosti, které byly identifikovány simulačně v teoretické části. Do systému vstupuje výrazná porucha díky tvorbě vzdušných vírů. U identifikace se postupovalo tak, jak bylo popsáno v kapitole 2. 5. 1.



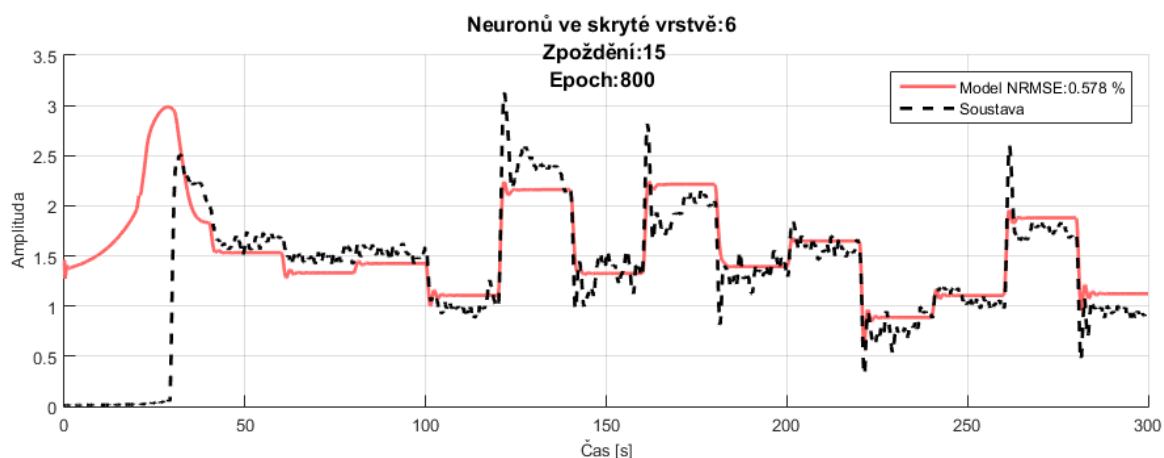
4-13 Identifikační signál pro model vrtulníku



4-14 Kontrolní signál pro model vrtulníku

Pro identifikaci byl použit signál chirp s namodulovaným bílým šumem. Výsledek vylepšilo přidání několika skoků, je však možné, že pouhým prodloužením chirp signálu v čase by bylo dosaženo podobného výsledku. Jelikož se pracovní oblast pohybuje ve velmi nízkém rozsahu hodnot napětí hlavního motoru, osvědčilo se neuronovou síť učit pouze v tomto rozsahu. V identifikačním signálu tedy chybí náběhová rampa do pracovní oblasti. Ta je zobrazená na obr 2-14, tedy kontrolním

signálu, ale pro ověření byla také vynechána. Jak na kontrolním signálu, tak i na identifikačním signálu je možné si všimnout vstupující poruchy, kde odezva systému na konstantní hodnotu nezanedbatelně osciluje.

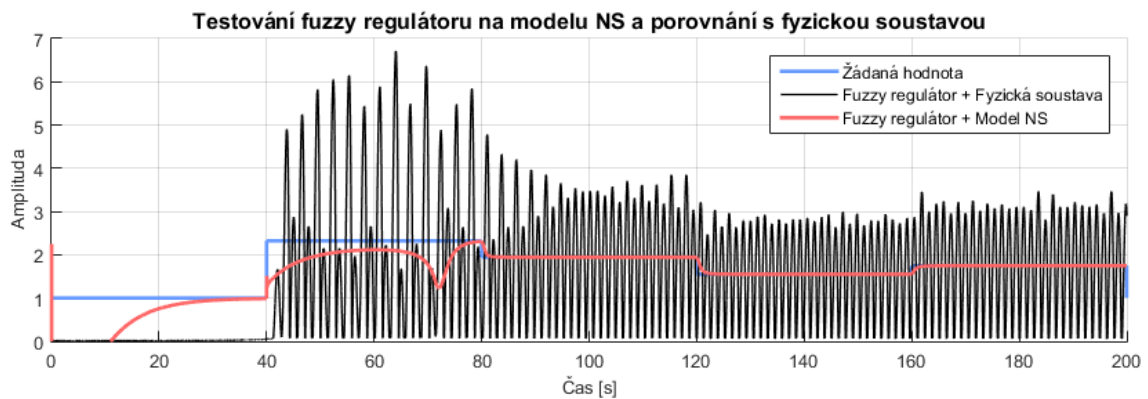


4-15 Identifikovaný model vrtulníku

Tabulka 5 NRMSE sítí s dobou učení 400 epoch

NRMSE		Neuronů ve skryté vrstvě					
		6	10	14	18	22	26
Zpoždění	6	-4,62	0,05	0,01	-25,60	-0,05	0,22
	9	-36,67	0,49	0,00	0,30	-4,88	-0,04
	12	-3,65	0,24	-2,28	-0,05	-0,01	0,10
	15	0,58	0,15	-3,79	0,12	-6,45	-9,52
	18	0,25	-0,10	-0,34	0,15	-0,05	0,22
	21	-5,85	0,14	0,41	-0,02	-5,84	-2,76
	24	0,23	0,20	-0,94	0,05	0,03	0,15

Z grafického výsledku je zřejmé, že neuronová síť s nejlepším výsledkem NRMSE má místy dobré výsledky, staticky se ustaluje na podobné hodnotě, ale nedokáže si poradit se vstupující poruchou. V tabulce je vidět, že většina sítí se k žádanému výsledku nepřiblížila. Na ověření, zda je identifikovaný model dostatečný pro realizaci regulátoru, byl na model simulačně navrhnout PI regulátor (obr. 4-16, zobrazen červeně). Po ověření jeho funkčnosti i na fyzickém systému bylo rozhodnuto, že na rozdíl od předchozí úlohy, bude fuzzy regulátor nastaven přímo na dynamické soustavě. Identifikovaný model systému není dostatečně přesný, aby popsal chování soustavy.



4-16 Navrhnutý regulátor na simulačním modelu a ověření funkčnosti na fyzickém systému

Jak bylo popsáno již dříve, zvyšováním napětí na hlavním motoru až do výstupního napětí přibližně 9 V není tah hlavního motoru dostatečný na to, aby se vrtulník začal vznášet. Naopak překročí-li napětí přibližně 9,3 V vrtulník se překlápí na zadní část, což není žádané. Je tedy zřejmé, že regulace bude probíhat ve velice malém pracovním rozsahu akční veličiny. Byl navržen regulátor typu fuzzy PID, do kterého navíc vstupuje informace o velikosti integračního výstupu regulátoru. Díky informaci o výstupním napětí je možné zvýšit přírůstek integrace v počátku, kdy je nutné překročit takové napětí, aby tah motoru začal zvedat váhu vrtulníku. Tím je dosaženo rychlejšího startu.

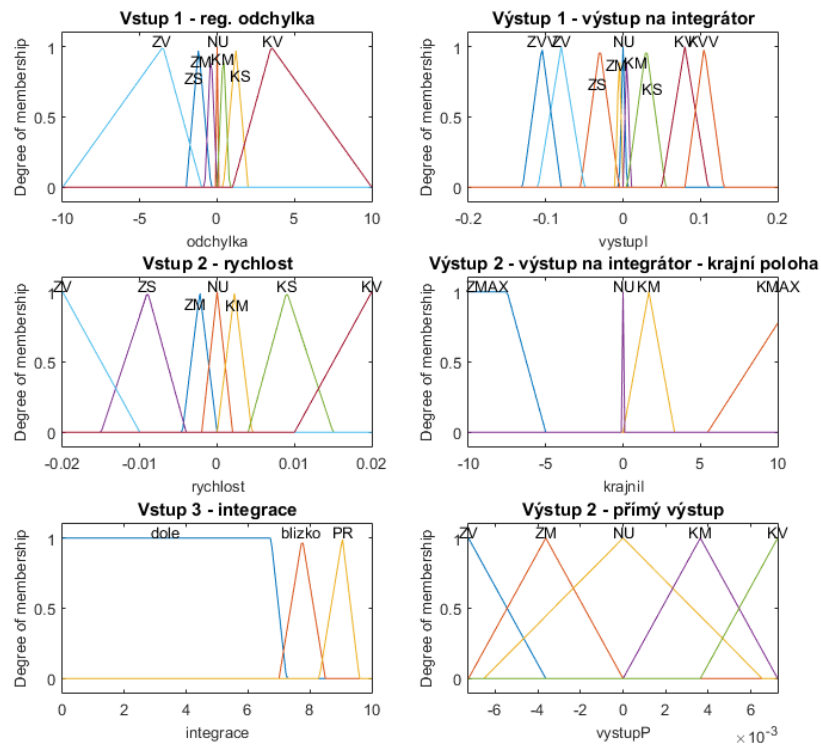
Dále bylo provedeno další měření, kde byly orientačně určeny velikosti lingvistické hodnoty „rychlost“. Při vypnutých motorech byl vrtulník naklápěn, a to takovou rychlostí, jaká se řešiteli zdála odpovídající velikosti 'malá', 'střední' a 'velká'. Poté byly empiricky navrženy fuzzy hodnoty vstupní proměnné „odchylka“ a výstupní proměnné „vystupI“ a „vystupP“. Tyto čtyři proměnné byly postupně upravovány tak, aby regulátor potlačil kmitání, a přitom reagoval stále dostatečně pohotově. Dále následoval návrh vstupní hodnoty „integrace“, který vyplýval z pracovního rozsahu soustavy.

Výstupní funkce, vstupující na integrátor, byla kvůli přehlednosti rozdělena na dvě, které se sečtou. Jelikož výstupní funkce „krajní“, zajišťující rychlý start, je hodnotami řádově výše než výstup v oblasti, kdy je vrtulník již nadnášen tahem vrtule, grafické zobrazení Matlabu se stalo v oblasti malých hodnot nepřehledné.

Je-li hodnota „integrace“ v oblasti „dole“ projeví se výstup „krajní“ hodnotou „KMAX“. Dostane-li se hodnota „integrace“ do oblasti „blízko“ omezí se výstup hodnoty „krajní“ na hodnotu „KM“. V oblasti „integrace“ na hodnotách „PR“ je pak

hodnota výstupu „krajni“ nulová a probíhá regulace na výstupu „vystup“ v závislosti na vstupních hodnotách „odchylka“ a „rychlost“.

Níže, na obrázku 4-17 lze vidět grafické znázornění funkcí příslušnosti nastavení fuzzy regulátoru navržených pro hlavní motor. Tabulka 6 graficky zobrazuje bázi inferenčních pravidel v pracovní oblasti. Pravidla jsou navržena dle známých metaprávidel (viz [2]) a upravena na základě empirických znalostí chování soustavy. Příloha zahrnuje funkce příslušnosti rozepsané matematicky a Příloha B přehled inferenčních pravidel.

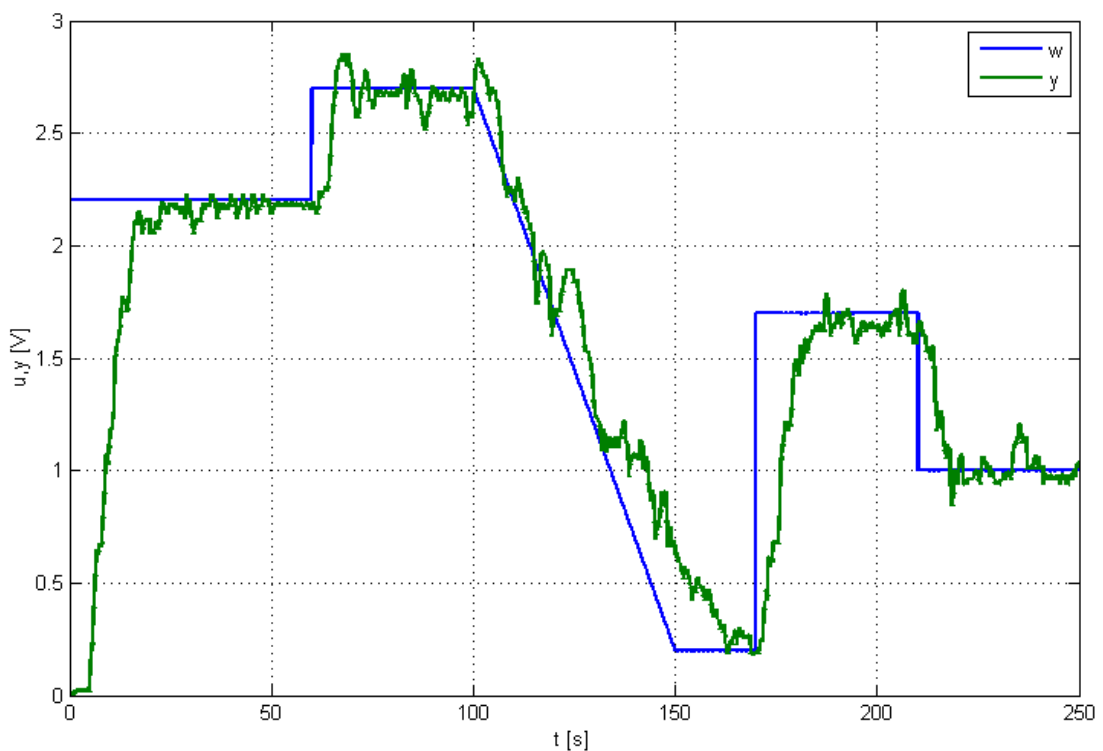


4-17 Funkce příslušnosti fuzzy regulátoru pro hlavní motor modelu

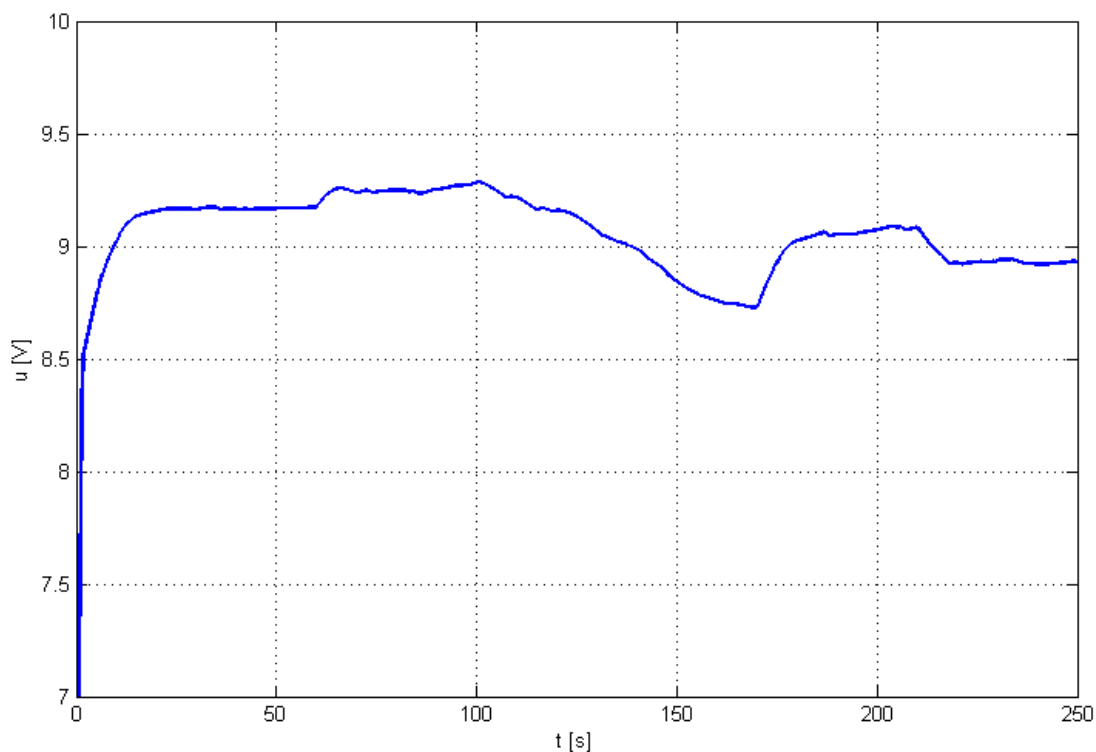
Tabulka 6 Inferenční pravidla v pracovní oblasti

		rychlost						
		ZV	ZS	ZM	NU	KM	KS	KV
odchylka	ZV	ZVV	ZV	ZV	ZV	ZS	NU	KM
	ZS	ZVV	ZS	ZS	ZS	ZM	NU	
	ZM	ZVV	ZS	ZM	ZM	NU	KM	
	NU	ZVV	ZM	ZM	NU	KM	KM	KVV
	KM		ZM	NU	KM	KM	KS	KVV
	KS	ZS	NU	KM	KS	KS	KS	KVV
	KV	ZM	NU	KS	KV	KV	KV	KVV

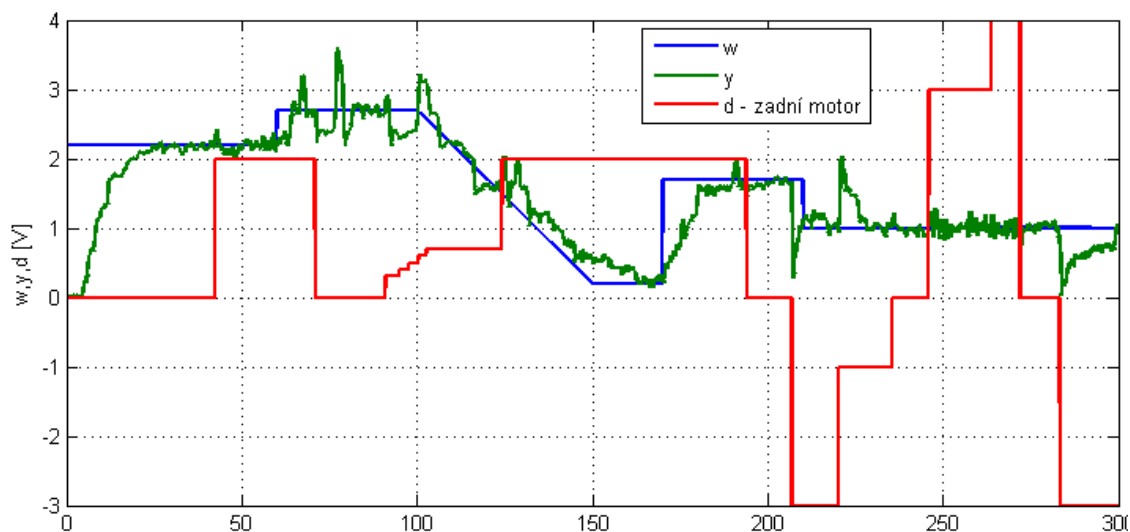
Na obrázku 4-18 je výsledek regulace. Modrou barvou je vyznačena žádaná hodnota, zelenou barvou regulovaná veličina. Je možné si všimnout kmitání okolo žádané hodnoty způsobené vzdušnými víry. To lze vidět na obrázku 4-19, kde je zobrazen akční zásah regulátoru, který se po ustálení mění minimálně. Díky zakomponování třetí vstupní proměnné do fuzzy regulátoru si lze všimnout, jak rychle regulátor dosáhl vysokého akčního zásahu. Na obrázku 4-20 byl během regulace spínán zadní motor (vyznačeno červenou barvou), který v této konfiguraci představuje další variantu dodané poruchy. Je vidět, že moment vyvíjený při změny rychlosti otáčení zadního motoru ovlivňuje naklonění vrtulníku.



4-18 Fuzzy regulace v uzavřené smyčce

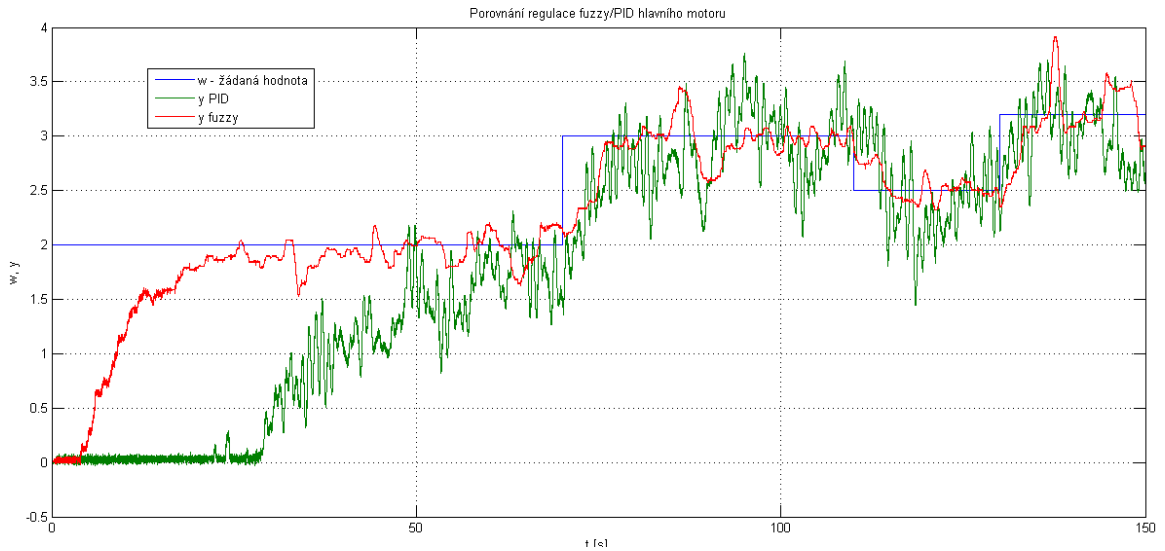


4-19 Akční zásah fuzzy regulátoru



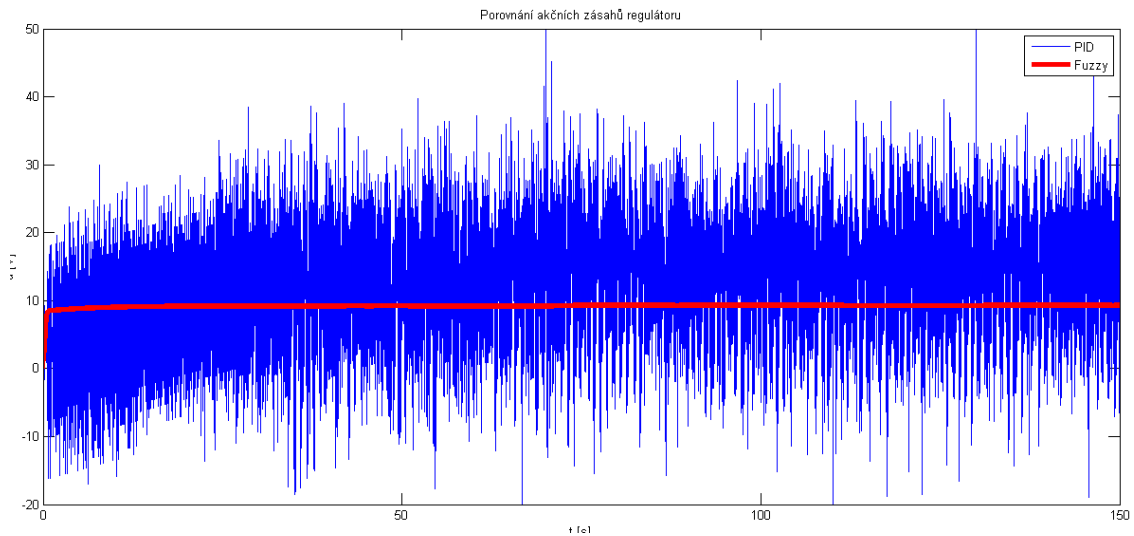
4-20 Regulace v uzavřené smyčce s vazbou od zadního motoru

Na obrázku 4-21 je porovnání regulované veličiny fuzzy regulátoru a PID regulátoru experimentálně navrženém v laboratoři. Na první pohled je zřejmý rychlejší start fuzzy regulátoru, díky zvýšenému akčnímu zásahu při nízké hodnotě naintegrované na akční veličině. Dále je průběh fuzzy regulátoru oproti PID regulátoru hladší a méně kmitavý. Odchylky od žádané hodnoty jsou způsobeny vzdušnými víry vytvářených rotací vrtule.



4-21 Porovnání fuzzy regulátoru a PID regulátoru

Na obrázku 4-22 je porovnání akční veličiny fuzzy a experimentálního PID regulátoru. Akční veličina fuzzy regulátoru je výrazně hladší, nastavení PID regulátoru v laboratoři je velice kmitavé a často se nachází nad úrovní saturace (0–10 V) akčního členu.



4-22 Porovnání akční veličiny fuzzy regulátoru a akční veličiny PID regulátoru

5. Závěr

Z výsledků fuzzy řízení lze usoudit, že je možné dosáhnout velice dobrých výsledků, s podobnou časovou náročností jako u konvenčního PID přístupu. U modelu vrtulníku fuzzy regulátor dokázal, oproti PID regulátoru experimentálně navrženém v laboratoři, mimo jiné výrazně urychlit start vrtulníku. Start vrtulníku řízeného PID regulátorem je navíc ovlivněn velikostí žádané hodnoty, respektive regulační odchylky. Pokud bude regulační odchylka malá, bude doba potřebná pro integraci napětí dostatečným pro start vzrůstat. Zavedením třetí vstupní proměnné u fuzzy regulátoru je tento nedostatek potlačen. Po startu byl průběh fuzzy regulátoru oproti PID regulátoru méně kmitavý. Akční zásahy fuzzy regulátoru jsou též méně kolísavé, nedocházelo by tedy v některých aplikacích k nadměrnému namáhání akčního členu. Na podobně složité soustavy (MIMO, nelineární či systémy s neočekávanou poruchou) má tedy nasazení fuzzy regulátoru jistě opodstatnění a lze tak díky variabilitě parametrů uspokojit požadavky na regulaci. Naopak expertní přístup vede často na metodu pokus-omyl, která může být zdlouhavá, v některých případech i nemožná (kvůli poškození soustavy, či bezpečnosti). Pro jednoduché soustavy nebo pro soustavy s menšími nároky na kvalitu regulačního pochodu není aplikace fuzzy regulátoru příliš opodstatněná.

Podobně neuronové sítě lze považovat za perspektivní metodu k identifikaci dynamických systémů. V teoretické části bylo dokázáno, že je možné neuronovou síť použít na různé typy soustav. V této části práce byly také diskutovány možné parametry učení a jejich dopad na výsledek učení. Dále bylo popsáno, jaký identifikační signál je vhodné volit. Osvědčil se signál, který obsáhne celou pracovní oblast, ideálně zasahuje mírně přes pracovní oblast, s namodulovaným bílým šumem a periodickou složkou, například ve formě sinusového či chirp signálu.

V praktické části byla ověřena funkčnost i na reálných soustavách. I zde lze identifikaci v případě prvních dvou soustav úspěšnou, v případě vrtulníku jako zdařilou (výsledek částečně dehonestuje zvolený klasifikátor NRMSE a počáteční inicializace, avšak grafickým porovnáním, s přihlédnutím na nestabilitu soustavy, kde je vstupující porucha nepředvídatelná a prakticky i neopakovatelná, je výsledek více než obstojný). Neuronové sítě se také osvědčily k prvotnímu nastavení regulátoru, je však vhodné výsledek doladit na fyzické soustavě.

Jak fuzzy řízení, tak neuronové sítě, těžší a zároveň trpí oproti konvenčnímu přístupu na společnou skutečnost, a to je velké množství parametrů. U fuzzy řízení tento fakt přidává na větší možnosti zlepšení a adaptace na požadavky procesu. Neuronové sítě často vedou na heuristický přístup. Správná volba všech parametrů je stále předmětem zkoumání. Díky prvku náhody na začátku učení je tak možné, že sít s parametry, které podaly výborné výsledky, poskytnou po zopakování učení nepoužitelný výkon.

V práci si by se tak dalo pokračovat například hlubším testováním vlivu parametrů, či hledáním správné metody inicializace zpožděvacích členů v simulaci, díky kterým se sítě v prvních časových okamžicích simulace nechovaly dle očekávání, i přes to, že byl výsledek po odeznění tohoto děje více než dobrý. Dále hledání předpokládání před učením, tak aby nedocházelo k rozporuplným výsledkům pro každý cyklus učení. Pokud by tohoto bylo dosaženo, lze dále hledat korelaci mezi potřebnými parametry neuronové sítě k popisu soustavy v závislosti na její složitosti. Naopak došlo by k předpokládání, zničil by rozpor v principech softcomutingu, který je na existenci náhody založen.

Použitá literatura

- [1] NOVÁK, Vilém. Základy fuzzy modelování. 1. vyd. Praha: BEN - technická literatura, 2000, 175 s. ISBN 80-7300-009-1.
- [2] Modrlák O., Fuzzy řízení a regulace – Teorie automatického řízení II, Fakulta mechatroniky a mezioborových inženýrských studií, 2002, 25 s. Skripta. Technická univerzita v Liberci
- [3] Fuzzy Logic and its Practical use in Mass Transit Systems. [online]. [cit. 2. 3. 2015]. URL: <http://skisko.blogspot.cz/2005/06/fuzzy-logic-and-its-practical-use-in.html>
- [4] GEBHARDT, Jörg. Fuzzy Application Library-Technical Applications-Industrial Automation. [online]. [cit. 2. 3. 2015]. URL: http://www.fuzzytech.com/e/e_a_plc.html
- [5] KLINGENBERG, Bryan. Fuzzy Logic Applications. [online]. [cit. 2. 3. 2015]. URL: <https://www.calvin.edu/~pribeiro/othrlnks/Fuzzy/apps.htm>
- [6] YASUNOBU, Seiji a MIYAMOTO. Automatic train operation system by predictive fuzzy control. [online]. [cit. 2. 3. 2015]. URL: <http://www.ics.esys.tsukuba.ac.jp/yasunobu/papers/ATO1985.pdf>
- [7] The Foundations of Fuzzy Control. [online]. [cit. 2. 3. 2015]. URL: https://books.google.cz/books?id=4zRKARW3-h8C&pg=PA130&lpg=PA130&dq=fuzzy+subway+sendai&source=bl&ots=QdO660AnBF&sig=2LUu57mEl_fDMqF4RfOv5yAeWPA&hl=cs&sa=X&ei=FNfsVJ-RLIm7ygOOQiYDoCw&ved=0CG0Q6AEwCw#v=onepage&q=fuzzy%20subway%20sendai&f=false
- [8] SEDONÍK, Jiří. Implementace báze pravidel a fuzzy regulátoru v rozhodovacích problémech demografických studií [online]. Olomouc, 2012 [cit. 5. 1. 2015]. URL: http://theses.cz/id/sqtk22/Sedonik_DP.pdf. Magisterská práce. Univerzita Palackého v Olomouci Přírodovědecká fakulta Katedra geoinformatiky. Vedoucí práce Mgr. Pavel Tuček, Ph.D.
- [9] Fuzzy Application Library-Technical Applications-Antilock-Braking System: Antilock-Braking System and Vehicle Speed Estimation using Fuzzy Logic [online]. 1996. Paris [cit. 26. 2. 2015]. URL: <http://www.fuzzytech.com/>
- [10] NEDELJKOVIC, I. Image classification based on fuzzy logic [online]. [cit. 26. 2. 2015]. URL: <http://www.isprs.org/proceedings/XXXV/congress/comm6/papers/685.pdf>
- [11] JADON, Brajpal Singh. 2013. Fuzzy Logic Technique in Digital Images using Edge Detection [online]. [cit. 26. 2. 2015]. URL: <https://www.rgpv.ac.in/iccbdt/papers/cl-143.pdf>
- [12] MASSIVE (software). 2011. Wikipedia [online]. [cit. 26. 2. 2015]. URL: http://en.wikipedia.org/wiki/MASSIVE_%28software%29
- [13] Products/fuzzyTECH Editions/Features Overview. FuzzyTECH [online]. [cit. 30. 2. 2015]. URL: <http://www.fuzzytech.com/>
- [14] Fuzzylite: a fuzzy logic control library and application [online]. 2014. [cit. 30. 2. 2015]. URL: <http://www.fuzzylite.com/>
- [15] OMRON SYSMAC C200H-FZ001 operational manual: Fuzzy Logic Unit [online]. 1992. [cit. 30. 2. 2015]. URL: http://omronkft.hu/nostree/pdfs/plc/c200h/w208e12_c200h-fz001_operation_manual.pdf
- [16] FuzzyControl++: Configuration tool for fuzzy logic. SIEMENS [online]. [cit. 30. 2. 2015]. URL: <https://mall.industry.siemens.com/mall/en/de/Catalog/Products/10020318>
- [17] BISKUP, Roman. Možnosti neuronových sítí. Praha, 2009. Disertační práce. Česká zemědělská univerzita v Praze. Vedoucí práce Prof. RNDr. Anna Čermáková, CSc.
- [18] VONDRÁK, CSC., prof. Ing. Ivo. Neuronové sítě. Ostrava, 2009. Technická univerzita Ostrava.
- [19] VOLNÁ, EVA. Neuronové sítě 1. Ostrava, 2008. Ostravská univerzita v Ostravě.
- [20] HAGAN, Martin T. Neural Network Design: 2nd Edition. In: <http://go.okstate.edu/> [online]. [cit. 2016-10-26]. Dostupné z: <http://hagan.okstate.edu/NNDesign.pdf>

- [21] ROJAS, Raúl. Neural Networks: A Systematic Introduction [online]. In: . Berlin, 1996 [cit. 2016-10-26]. Dostupné z: <http://page.mi.fu-berlin.de/rojas/neural/neuron.pdf>
- [22] EGMONT-PETERSEN, M., D. DE RIDDER a H. HANDELS. Image processing with neural networks—a review [online]. Amsterdam, 2001 [cit. 2016-11-04]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.8958&rep=rep1&type=pdf>
- [23] A. EL-SAYED, Mohamed, Yarub A. ESTAITIA a Mohamed A. KHAFAGY. Automated Edge Detection Using Convolutional Neural Network [online]. Bradford (UK), 2013 [cit. 2016-11-04]. Dostupné z: http://thesai.org/Downloads/Volume4No10/Paper_3-Automated_Edge_Detection_Using_Convolutional.pdf
- [24] MALTAROLLO, ViníciusGonçalves, Káthia Maria HONÓRIO a Albérico Borges Ferreira DA SILVA. Applications of Artificial Neural Networks in Chemical Problems [online]. InTech, 2013 [cit. 2016-11-04]. Dostupné z: <http://www.intechopen.com/books/artificial-neural-networks-architectures-and-applications/applications-of-artificial-neural-networks-in-chemical-problems>
- [25] SPIWOK, Vojtěch a Blanka KRÁLOVÁ. Virtuální screening léčiv [online]. Praha, 2008 [cit. 2016-11-04]. Dostupné z: http://www.chemicke-listy.cz/docs/full/2009_01_52-55.pdf
- [26] DEAN A, Pomerleau. ALVINN, anautonomous land vehicle in a neural network [online]. Carnegie Mellon University, 1989 [cit. 2016-11-04]. Dostupné z: <http://repository.cmu.edu/cgi/viewcontent.cgi?article=2874&context=compsci>
- [27] DEAN A, Pomerleau. Neural Network Vision for Robot Driving [online]. Carnegie Mellon University [cit. 2016-11-04]. Dostupné z: https://www.ri.cmu.edu/pub_files/pub2/pomerleau_dean_1995_1/pomerleau_dean_1995_1.pdf
- [28] LEE, Saro a Hyun-Joo OH. Application of Artificial Neural Network for Mineral Potential Mapping [online]. Republic of Korea, 2011 [cit. 2016-11-07]. Dostupné z: <http://www.intechopen.com/books/howtoreference/artificial-neural-networks-application/application-of-artificial-neural-network-for-mineral-potential-mapping>
- [29] ŠTENCL, Michael a ŠŤASTNÝ. Artificial Neural Networks Numerical Forecasting of Economic Time Series [online]. 2011 [cit. 2016-11-07]. Dostupné z: <http://www.intechopen.com/books/howtoreference/artificial-neural-networks-application/artificial-neural-networks-numerical-forecasting-of-economic-time-series>
- [30] LATHA, P., L. GANESAN a S. ANNADURAI. Face RecognitionusingNeuralNetworks [online]. [cit. 2016-11-07]. Dostupné z: <http://www.cscjournals.org/manuscript/Journals/SPIJ/Volume3/Issue5/SPIJ-37.pdf>
- [31] KOLASA – WIĘCEK, Alicja. Application of Artificial Intelligence in Environmental Sciences – Forecasting CO 2 Emission in Poland [online]. Opole (Poland), 2011 [cit. 2016-11-07]. Dostupné z: <http://www.intechopen.com/books/artificial-neural-networks-application/application-of-artificial-intelligence-in-environmental-sciences-forecasting-co2-emission-in-poland>
- [32] MODRLÁK, Osvald a Lukáš HUBKA. Otáčky DC motoru – „DC motor se zátěží“. Liberec, 2010.
- [33] MODRLÁK, Osvald a Lukáš HUBKA. Teplota proudícího vzduchu – „Fén“. Liberec, 2010.
- [34] KARER, G. a B. ZUPANČIČ. Modelling and identifiaion of a laboratory helicopter. Ljubljana.
- [35] PRECHELT, Lutz. Early Stopping | but when? [online]. Karlsruhe, Germany [cit. 2016-11-17]. Dostupné z: http://page.mi.fu-berlin.de/prechelt/Biblio/stop_tricks1997.pdf
- [36] Vícevrstvá neuronová síť. In: Matematická biologie [online]. [cit. 2016-12-05]. Dostupné z: <http://portal.matematickabiologie.cz/res/image/Umela%20intelligence/obr-4-4-usporadani-neuronu-do-vrstev.png>
- [37] NARX síť v softwaru MATLAB. In: Design Time Series NARX Feedback NeuralNetworks [online]. [cit. 2016-12-05]. Dostupné z: https://www.mathworks.com/help/examples/nnet/UgTsNarxFeedbackExample_03.png

- [38] Vrtulník CE 150 [online]. In: . [cit. 2016-12-05]. Dostupné z:
http://images2.hellotrade.com/data3/IQ/FA/HTVENDOR-7370538/produkty-models-ce150-heli_ph-250x250.gif
- [39] Biologický neuron [online]. In: . [cit. 2016-12-08]. Dostupné z:
<http://media1.7x.cz/images/media1:50fee219bb85e.jpg/Neuron-cs.jpg>
- [40] Model perceptronu. In: Elektrorevue.cz [online]. [cit. 2016-12-08]. Dostupné z:
<http://www.elektrorevue.cz/clanky/00013/Image3.gif>
- [41] Elektrorevue.cz [online]. [cit. 2016-12-08]. Dostupné z:
<http://www.elektrorevue.cz/clanky/00013/Image3.gif>

Přílohy

Příloha A Funkce příslušnosti fuzzy regulátoru vrtulníku

```
[Input1]
name: 'odchylka'
range: [-10 10]
'ZS'    'trimf'  [-2 -1,2 -0,4]
'NU'    'trimf'  [-0,05 0 0,05]
'KS'    'trimf'  [0,4 1,2 2]
'ZM'    'trimf'  [-0,8 -0,4 0]
'KM'    'trimf'  [0 0,4 0,8]
'ZV'    'trimf'  [-10 -3,5 -1]
'KV'    'trimf'  [1 3,5 10]

[Input2]
name: 'rychlost'
range: [-0.0200 0.0200]
'ZM'    'trimf'  [-0,0045 -0,00225 0]
'NU'    'trimf'  [-0,002 0 0,002]
'KM'    'trimf'  [0 0,00225 0,0045]
'ZS'    'trimf'  [-0,015 -0,009 -0,004]
'KS'    'trimf'  [0,004 0,009 0,015]
'ZV'    'trapmf' [-Inf -Inf -0,02 -0,01]
'KV'    'trapmf' [0,01 0,02 InfInf]

[Input3]
name: 'integrace'
range: [0 10]
'dole'  'trapmf' [-0,265 -0,265 6,74 7,24]
'blizko' 'trimf' [7 7,75 8,5]
'PR'    'trimf' [8,3 9,05 9,6]

[Output1]
name: 'vystupI'
range: [-0.2000 0.2000]
'NU'    'trimf'  [-0,005 0 0,005]
'ZS'    'trimf'  [-0,055 -0,03 -0,005]
'ZM'    'trimf'  [-0,01 -0,005 0]
'KM'    'trimf'  [0 0,005 0,01]
'KS'    'trimf'  [0,005 0,03 0,055]
'ZV'    'trimf'  [-0,11 -0,08 -0,05]
'KV'    'trimf'  [0,05 0,08 0,11]
'ZVV'   'trimf'  [-0,13 -0,105 -0,08]
'KVV'   'trimf'  [0,08 0,105 0,13]

[Output2]
name: 'krajnil'
range: [-10 10]
'ZMAX'  'trapmf' [-Inf -Inf -7,5 -5]
'KMAX'  'trapmf' [5,4773 11,3073 InfInf]
'KM'    'trimf'  [0 1,668 3,332]
'NU'    'trimf'  [-0,03334 0 0,03334]

[Output3]
name: 'vystupP'
range: [-0.0073 0.0073]
'ZV'    'trimf'  [-0,01087 -0,00725 -0,003625]
'ZM'    'trimf'  [-0,00725 -0,003625 0]
'NU'    'trimf'  [-0,006525 0 0,006525]
'KM'    'trimf'  [0 0,003625 0,00725]
'KV'    'trimf'  [0,003625 0,00725 0,01088]
```

Příloha B Inferenční pravidla fuzzy regulátoru vrtulníku

1. If (odchylka is KV) and (rychlost is KS) and (integrace is PR) then (vystuplis KV)(vystupPis KV) (1)
2. If (odchylka is KV) and (rychlost is KM) and (integrace is PR) then (vystuplis KV)(vystupPis KV) (1)
3. If (odchylka is KV) and (rychlost is NU) and (integrace is PR) then (vystuplis KV)(vystupPis KV) (1)
4. If (odchylka is KV) and (rychlost is ZM) and (integrace is PR) then (vystuplis KS)(vystupPis KM) (1)
5. If (odchylka is KV) and (rychlost is ZS) and (integrace is PR) then (vystuplis NU)(vystupPis NU) (1)
6. If (odchylka is KS) and (rychlost is KS) and (integrace is PR) then (vystuplis KS)(vystupPis KM) (1)
7. If (odchylka is KS) and (rychlost is KM) and (integrace is PR) then (vystuplis KS)(vystupPis KM) (1)
8. If (odchylka is KS) and (rychlost is NU) and (integrace is PR) then (vystuplis KS)(vystupPis KM) (1)
9. If (odchylka is KS) and (rychlost is ZM) and (integrace is PR) then (vystuplis KM)(vystupPis KM) (1)
10. If (odchylka is KS) and (rychlost is ZS) and (integrace is PR) then (vystuplis NU)(vystupPis NU) (1)
11. If (odchylka is KM) and (rychlost is KS) and (integrace is PR) then (vystuplis KS)(vystupPis KM) (1)
12. If (odchylka is KM) and (rychlost is KM) and (integrace is PR) then (vystuplis KM)(vystupPis KM) (1)
13. If (odchylka is KM) and (rychlost is NU) and (integrace is PR) then (vystuplis KM)(vystupPis KM) (1)
14. If (odchylka is KM) and (rychlost is ZM) and (integrace is PR) then (vystuplis NU)(vystupPis NU) (1)
15. If (odchylka is KM) and (rychlost is ZS) and (integrace is PR) then (vystuplis ZM)(vystupPis ZM) (1)
16. If (odchylka is NU) and (rychlost is KS) and (integrace is PR) then (vystuplis KM)(vystupPis KM) (1)
17. If (odchylka is NU) and (rychlost is KM) and (integrace is PR) then (vystuplis KM)(vystupPis KM) (1)
18. If (odchylka is NU) and (rychlost is NU) and (integrace is PR) then (vystuplis NU)(vystupPis NU) (1)
19. If (odchylka is NU) and (rychlost is ZM) and (integrace is PR) then (vystuplis ZM)(vystupPis ZM) (1)
20. If (odchylka is NU) and (rychlost is ZS) and (integrace is PR) then (vystuplis ZM)(vystupPis ZM) (1)
21. If (odchylka is ZM) and (rychlost is KS) and (integrace is PR) then (vystuplis KS)(vystupPis KM) (1)
22. If (odchylka is ZM) and (rychlost is KM) and (integrace is PR) then (vystuplis NU)(vystupPis NU) (1)
23. If (odchylka is ZM) and (rychlost is NU) and (integrace is PR) then (vystuplis ZM)(vystupPis ZM) (1)
24. If (odchylka is ZM) and (rychlost is ZM) and (integrace is PR) then (vystuplis ZS)(vystupPis ZM) (1)
25. If (odchylka is ZM) and (rychlost is ZS) and (integrace is PR) then (vystuplis ZS)(vystupPis ZM) (1)
26. If (odchylka is ZS) and (rychlost is KS) and (integrace is PR) then (vystuplis NU)(vystupPis NU) (1)
27. If (odchylka is ZS) and (rychlost is KM) and (integrace is PR) then (vystuplis ZM)(vystupPis ZM) (1)
28. If (odchylka is ZS) and (rychlost is NU) and (integrace is PR) then (vystuplis ZS)(vystupPis ZM) (1)
29. If (odchylka is ZS) and (rychlost is ZM) and (integrace is PR) then (vystuplis ZS)(vystupPis ZM) (1)
30. If (odchylka is ZS) and (rychlost is ZS) and (integrace is PR) then (vystuplis ZS)(vystupPis ZM) (1)
31. If (odchylka is ZV) and (rychlost is KS) and (integrace is PR) then (vystuplis NU)(vystupPis NU) (1)
32. If (odchylka is ZV) and (rychlost is KM) and (integrace is PR) then (vystuplis ZS)(vystupPis ZM) (1)
33. If (odchylka is ZV) and (rychlost is NU) and (integrace is PR) then (vystuplis ZV)(vystupPis ZV) (1)
34. If (odchylka is ZV) and (rychlost is ZM) and (integrace is PR) then (vystuplis ZV)(vystupPis ZV) (1)
35. If (odchylka is ZV) and (rychlost is ZS) and (integrace is PR) then (vystuplis ZV)(vystupPis ZV) (1)
36. If (odchylka is KV) and (rychlost is ZV) and (integrace is PR) then (vystuplis ZM)(vystupPis ZM) (1)
37. If (odchylka is KS) and (rychlost is ZV) and (integrace is PR) then (vystuplis ZS)(vystupPis ZM) (1)
38. If (odchylka is ZV) and (rychlost is KV) and (integrace is PR) then (vystuplis KM)(vystupPis KM) (1)
39. If (odchylka is NU) and (rychlost is KV) and (integrace is PR) then (vystuplis KVV) (1)
40. If (odchylka is KM) and (rychlost is KV) and (integrace is PR) then (vystuplis KVV) (1)
41. If (odchylka is KS) and (rychlost is KV) and (integrace is PR) then (vystuplis KVV) (1)
42. If (odchylka is KV) and (rychlost is KV) and (integrace is PR) then (vystuplis KVV) (1)
43. If (odchylka is NU) and (rychlost is ZV) and (integrace is PR) then (vystuplis ZVV) (1)
44. If (odchylka is ZM) and (rychlost is ZV) and (integrace is PR) then (vystuplis ZVV) (1)
45. If (odchylka is ZS) and (rychlost is ZV) and (integrace is PR) then (vystuplis ZVV) (1)
46. If (odchylka is ZV) and (rychlost is ZV) and (integrace is PR) then (vystuplis ZVV) (1)
47. If (integrace is dole) then (vystuplis NU)(krajnilis KMAX) (1)
48. If (integrace is blizko) then (vystuplis NU)(krajnilis KM) (1)
49. If (integrace is PR) then (krajnilis NU) (1)

Příloha C Funkce příslušnosti fuzzy PI regulátoru motoru

```
[Input1]
name: 'e'
range: [-10 10]
'ZV'    'trimf'  [-15 -10 -5]
'ZM'    'trimf'  [-10 -5 0]
'NU'    'trimf'  [-5 0 5]
'KM'    'trimf'  [0 5 10]
'KV'    'trimf'  [5 10 15]

[Output1]
name: 'ul'
range: [-3.1 3.1]
'ZV'    'trimf'  [-4,650 -3,1-1,55]
'ZM'    'trimf'  [-2,756 -1,377 0]
'NU'    'trimf'  [-1,292 0 1,292]
'KM'    'trimf'  [0 1,378 2,756]
'KV'    'trimf'  [1,55 3,1 4,65]

[Output2]
name: 'uP'
range: [-1,21,2]
'ZV'    'trimf'  [-1,8 -1,2 -0,6]
'ZM'    'trimf'  [-1,2 -0,6 0]
'NU'    'trimf'  [-0,96 0 0,96]
'KM'    'trimf'  [0 0,6 1,2]
'KV'    'trimf'  [0,6 1,2 1,8]
```

Příloha D Inferenční pravidla fuzzy PI regulátoru motoru

1. If (e is ZV) then (ulis ZV)(uPis ZV) (1)
2. If (e is ZM) then (ulis ZM)(uPis ZM) (1)
3. If (e is NU) then (ulis NU)(uPis NU) (1)
4. If (e is KM) then (ulis KM)(uPis KM) (1)
5. If (e is KV) then (ulis KV)(uPis KV) (1)

Příloha E Funkce příslušnosti fuzzy PID regulátoru motoru

```
[Input1]
name: 'e'
range: [-10 10]
'zv'    'trimf'  [-15 -10 -5]
'zm'    'trimf'  [-10 -5 0]
'nu'    'trimf'  [-5 0 5]
'km'    'trimf'  [0 5 10]
'kv'    'trimf'  [5 10 15]

[Input2]
name: 'de'
range: [-6 6]
'zv'    'trimf'  [-9 -6 -3]
'zm'    'trimf'  [-6 -3 0]
'nu'    'trimf'  [-3 0 3]
'km'    'trimf'  [0 3 6]
'kv'    'trimf'  [3 6 9]

[Output1]
name: 'ul'
range: [-3.1 3.1]
'zv'    'trimf'  [-4,65 -3,1 -1,55]
'zm'    'trimf'  [-2,756 -1,377 0]
'nu'    'trimf'  [-1,292 0 1,292]
'km'    'trimf'  [0 1,378 2,756]
'kv'    'trimf'  [1,55 3,1 4,65]

[Output2]
name: 'uP'
range: [-1.2 1.2]
'zv'    'trimf'  [-1,8 -1,2 -0,6]
'zm'    'trimf'  [-1,2 -0,6 0]
'nu'    'trimf'  [-0,96 0 0,96]
'km'    'trimf'  [0 0,6 1,2]
'kv'    'trimf'  [0,6 1,2 1,8]
```

Příloha F Inferenční pravidla fuzzy PID regulátoru motoru

1. If (e iszv) and (de iszv) then (ulis nu)(uPis nu) (1)
2. If (e iszv) and (de iszm) then (ulism)(uPism) (1)
3. If (e iszv) and (de is nu) then (uliszv)(uPiszv) (1)
4. If (e iszm) and (de iszv) then (ulis km)(uPis km) (1)
5. If (e iszm) and (de iszm) then (ulis nu)(uPis nu) (1)
6. If (e iszm) and (de is nu) then (ulism)(uPism) (1)
7. If (e iszm) and (de is km) then (uliszv)(uPiszv) (1)
8. If (e is nu) and (de iszv) then (uliskv)(uPiskv) (1)
9. If (e is nu) and (de iszm) then (ulis km)(uPis km) (1)
10. If (e is nu) and (de is nu) then (ulis nu)(uPis nu) (1)
11. If (e is nu) and (de is km) then (ulism)(uPism) (1)
12. If (e is nu) and (de iskv) then (uliszv)(uPiszv) (1)
13. If (e is km) and (de iszm) then (uliskv)(uPiskv) (1)
14. If (e is km) and (de is nu) then (ulis km)(uPis km) (1)
15. If (e is km) and (de is km) then (ulis nu)(uPis nu) (1)
16. If (e is km) and (de iskv) then (ulism)(uPism) (1)
17. If (e iskv) and (de is nu) then (uliskv)(uPiskv) (1)
18. If (e iskv) and (de is km) then (ulis km)(uPis km) (1)
19. If (e iskv) and (de iskv) then (ulis nu)(uPis nu) (1)

Příloha G Program určený na učení sítí

```
%% Program určený pro identifikaci modelu neuronovou sítí
clear all
close all
clc;

% Soubor do které se uloží výsledky
storeFile = 'surfAB.mat';

%% Soustava
s = tf('s');
% P = [-4 -16];
P = [-4+8i -4-8i -6+30i -6-30i -20 -16];
Z = -8;
K = 2000000;
sys = zpkm(Z,P,K);
clear P Z K s

%% Identifikační signál a výpočet odezvy systému
maxT = 100;
ts = 0.05;
t = 0:ts:maxT;
x = [5*sin(t/(maxT/pi)) + 2*sin(t) + 1.5*rand(length(t),1)' + 1.5]';
y = lsim(sys,x,t);

%% Kontrolní signál
skoku = 10;
delka = 80;
tCh = 0:ts:delka;
ch1 = zeros(length(tCh),1);
inter = delka/skoku/ts;

for i = 1:skoku
    ch1((inter*(i-1)+1):inter*i) = 10*rand;
end
ch2 = lsim(sys,ch1,tCh);

%% Grafy signálů a odezev
if true
    actFig = figure;
    plot(t,x,t,y,'LineWidth',2);
    grid on
    title('Identifikační signál', 'FontSize', 12);
    legend('Vstupní signál','Odezva soustavy');
    xlabel('Čas [s]', 'FontSize', 10);
    ylabel('Amplituda', 'FontSize', 10);
    actFig.Color = [1 1 1];
    actFig = figure;
    plot(tCh,ch1,tCh,ch2,'LineWidth',2);
    grid on
    title('Kontrolní signál', 'FontSize', 12);
    legend('Vstupní signál','Odezva soustavy');
    xlabel('Čas [s]', 'FontSize', 10);
    ylabel('Amplituda', 'FontSize', 10);
    actFig.Color = [1 1 1];
end

%% Počet iterací přes parametry sítě
sizeA = 10; % zpoždění
sizeB = 10; % neuronů ve skryté vrstvě
sizeC = 1; % epoch

%% Přírůstek v další iteraci
incA = 1;
incB = 1;

%% První iterace (nejmenší síť s nejmenším počtem epoch)
startA = 4;
startB = 2;
startC = 1;

%% Načtení předchozích výsledků (pokud existují)
% Matice done označuje iterace, které již byly spočítané, lze tak
% vypočítávat síť s přerušením bez úplné ztráty informace
nets = num2cell (zeros(sizeA,sizeB,sizeC),ones(sizeA,1),ones(sizeB,1),ones(sizeC,1));
```

```

done = num2cell (zeros(sizeA,sizeB,sizeC),ones(sizeA,1),ones(sizeB,1),ones(sizeC,1));
if exist(storeFile,'file')
    load(storeFile);
    sizeA = size(nets,1);
    sizeB = size(nets,2);
    sizeC = size(nets,3);

    startA = nets{1,1,1}.numInputDelays;
    startB = nets{1,1,1}.layers{1}.dimensions;
    startC = nets{1,1,1}.trainParam.epochs;

    incA = nets{2,1,1}.numInputDelays - nets{1,1,1}.numInputDelays;
    incB = nets{1,2,1}.layers{1}.dimensions - nets{1,1,1}.layers{1}.dimensions;
else
    save(storeFile,'nets','done');
end

%% připravení dat na učení
X = tonndata(x,false,false);
T = tonndata(y,false,false);

%% Cyklus učení na identifikačním signálu
if false
for i=1:sizeA
    for j=1:sizeB
        for k=1:sizeC
            % Přeskoč pokud již bylo spočítáno
            if done{i,j,k}==true
                continue;
            end
            s = {strcat('Neurons in hidden layer: ',num2str(startB+((j-1)*incB)));strcat('Number of Delays: ',num2str(startA+((i-1)*incA)));strcat('Epochs: ',num2str((2^((k-1)+startC))*100))};
            disp(s);
            trainFcn = 'trainbr';
            inputDelays = 1:((i-1)*incA) +startA;
            feedbackDelays = 1:((i-1)*incA) + startA;
            hiddenLayerSize = startB+((j-1)*incB);

            net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize,'open',trainFcn);
            [x,xi,ai,t] = preparets(net,X,{},T);
            net.divideFcn = 'dividerand'; % Divide data randomly
            net.divideMode = 'time'; % Divide up every sample
            net.divideParam.trainRatio = 70/100;
            net.divideParam.valRatio = 15/100;
            net.divideParam.testRatio = 15/100;
            net.trainParam.epochs = (2^((k-1)+startC))*100;
            net.performFcn = 'mse'; % Mean Squared Error

            %% Trénování, uzavření smyčky, zapsání na HDD
            [net,tr] = train(net,x,t,xi,ai);
            nets{i,j,k}=net;
            netc = closeloop(net);
            done{i,j,k} = true;
            save(storeFile,'nets','netc','done');
        end
    end
end
end

% Příprava dat na kontrolní signál
outs = num2cell(zeros((size(nets,1)),size(nets,2),size(nets,3)));
mseNet = zeros((size(nets,1)),size(nets,2),size(nets,3));
testIn = num2cell(ch1);

%% Odezva z neuronové sítě
if true
    for i=1:sizeA
        for j=1:sizeB
            for k = 1:sizeC
                outs{i,j,k} = cnets{i,j,k}(testIn');
            end
        end
    end
end
end

```

```

%% Výpočet NRMSE
if true
    for i=1:sizeA
        for j=1:sizeB
            for k = 1:sizeC
                matOut = cell2mat(outs{i,j,k});
                mseNet(i,j,k) = goodnessOfFit(matOut,ch2,'NRMSE');
            end
        end
    end
end

bestS = 0.0;
%% Zobrazení výsledků
if true
    for i=1:sizeA
        for j=1:sizeB
            for k = 1:sizeC

                if mseNet(i,j,k) < bestS
                    continue;
                end
                bestS = mseNet(i,j,k);
                bestI = i;
                bestJ = j;
                bestK = k;

                actFig = figure();
                set(actFig,'Color',[1 1 1]);
                hold on;
                pl =cell2mat(outs{i,j,k});
                plot(tCh,pl(1,:), 'LineWidth',2,'Color',[1 0.4 0.4]);
                plot(tCh,ch2,'LineWidth',2,'Color','k','LineStyle','--');
                hold off;
                s = {strcat('Neuronů ve skryté vrstvě:
',num2str(cnets{i,j,k}.layers{1}.dimensions));strcat(' Zpoždění:
',num2str(cnets{i,j,k}.numLayerDelays));strcat(' Epoch:
',num2str(cnets{i,j,k}.trainParam.epochs))};
                title(s, 'FontSize', 24);
                legend('Odezva přenosové funkce',strcat('Neuronová síť. NRMSE: ',
num2str(mseNet{2,2,2}(1),3), ' %'));
                legend(strcat('Neuronová síť NRMSE: ', num2str(mseNet(i,j,k),3), '
%'),'Odezva přenosové funkce');
                xlabel('Čas [s]', 'FontSize', 20);
                ylabel('Amplituda', 'FontSize', 20);
                grid on;
            end
        end
    end
end

% NRMSE do excel tabulky
if false
    xlsName = 'mse_modSISO.xls';
    xlswrite(xlsName,mseNet(:, :, bestK),1,'C3');
    xlswrite(xlsName,[startA:incA:startA+incA*(sizeA-1)],1,'B3');
    xlswrite(xlsName,[startB:incB:startB+incB*(sizeB-1)],1,'C2');
    xlswrite(xlsName,{'Zpoždění'},1,'A3');
    xlswrite(xlsName,{'Neuronů ve skryté vrstvě'},1,'C1');
    xlswrite(xlsName,{'NRMSE'},1,'A1');

    Excel = actxserver('Excel.Application');
    Workbook = Excel.Workbooks.Open(strcat(pwd,'\ ',xlsName));
    Range = Excel.Range(strcat('A3:A',num2str(sizeA+2)));
    Range.Select;
    Range.MergeCells=true;
    Range.VerticalAlignment = 2;

    Range = Excel.Range(strcat('C1:',char(sizeB+66),'1'));
    Range.Select;
    Range.MergeCells=true;
    Range.HorizontalAlignment = 3;

    Range = Excel.Range(strcat('A1:', 'B2'));
    Range.Select;
    Range.MergeCells=true;
    Range.HorizontalAlignment = 3;
end

```

```

Range.VerticalAlignment = 2;

Workbook.SaveAs(strcat(pwd, '\', xlsName))
Excel.Workbooks.Close
end

%% Zobrazení povrchu
if true
    for Cdim=1:sizeC
        surfFig = figure;

surf(startA:incA:(sizeA*incA)+1, startB+1:incB:(sizeB*incB)+1, mseNet(2:end, :, Cdim));
        set(surfFig, 'Color', [1 1 1]);
        xlabel('Počet neuronů ve skryté vrstvě');

        ylabel('Počet zpoždění');
        title({'Plocha NRMSE pro různé parametry'; strcat('Počet trénovacích epoch:
', num2str(cnets{1,1,Cdim}.trainParam.epochs))});
        zlabel('NRMSE');
        colorbar;
        grid on;
    end
end

```