

**Univerzita Pardubice  
Fakulta ekonomicko-správní  
Ústav systémového inženýrství a informatiky**

**Model řízení na bázi výpočetní inteligence**

**Bc. Jindřich Hegr**

**Diplomová práce  
2019**

Univerzita Pardubice  
Fakulta ekonomicko-správní  
Akademický rok: 2018/2019

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jindřich Hegr**  
Osobní číslo: **E16686**  
Studijní program: **N6209 Systémové inženýrství a informatika**  
Studijní obor: **Regionální a informační management**  
Název tématu: **Model řízení na bázi výpočetní inteligence**  
Zadávající katedra: **Ústav systémového inženýrství a informatiky**

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

Cílem práce je návrh modelu řízení na bázi výpočetní inteligence (Soft Computing, Computational Intelligence) pro výukový model CE151 od firmy Humusoft.

Práce bude obsahovat:

- popis výukového modelu a vybraných metod výpočetní inteligence z pohledu řízení;
- možnosti návrhů řízení modelu na bázi výpočetní inteligence;
- návrh vybraných modelů řízení na bázi výpočetní inteligence;
- porovnání modelů řízení.

Rozsah grafických prací:

Rozsah pracovní zprávy: cca 50 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

BALÁTĚ, Jaroslav. Automatické řízení. 2. přeprac. vyd. Praha: BEN - technická literatura, 2004. 663 s. ISBN 80-7300-148-9.  
KARBAN, Pavel. Výpočty a simulace v programech Matlab a Simulink. 1. vyd. Brno: Computer Press, 2006. 220 s. ISBN 80-251-1301-9.  
JURA, Pavel. Základy fuzzy logiky pro řízení a modelování. 1. vyd. Brno: VUTIUM, 2003. 132 s. ISBN 80-214-2261-0.  
RUSSEL, Stuart - NORVIG, Peter. Artificial Intelligence: A Modern Approach. 3rd ed. Upper Saddle River: Pearson Education, 2010. 1152 s. ISBN 0-13-604259-7.  
NILSSON, Nils J. Artificial Intelligence: A New Synthesis. 1st ed. San Francisco: Morgan Kaufmann Publishers, Inc., 1998. 513 s. ISBN 7-111-07438-6.  
POKORNÝ, Miroslav. Umělá inteligence v modelování a řízení. 1. vyd. Praha: BEN - technická literatura, 1996. 187 s. ISBN 80-901984-4-9.  
NGUYEN, Hung T. - WALKER, Elbert A. A first course in fuzzy logic. 3rd ed. Boca Raton: Chapman & Hall, 2006. 440 s. ISBN 1-58488-526-2.  
JANG, Jyh-Shing R. - SUN, Chuen-Tsai - MIZUTANI, Eiji. Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. 1st ed. Upper Saddle River: Prentice-Hall, 1997. 614 s. ISBN 0-13-261066-3.  
HAYKIN, Simon S. Neural networks: a comprehensive foundation. 2nd ed. Upper Saddle River: Prentice Hall, 1999. 842 s. ISBN 0-13-273350-1.

Vedoucí diplomové práce:

doc. Ing. Jiří Křupka, Ph.D.

Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce:

3. září 2018

Termín odevzdání diplomové práce:

30. dubna 2019

doc. Ing. Romana Provažníková, Ph.D.  
děkanka

L.S.

doc. Ing. Pavel Petr, Ph.D.  
vedoucí ústavu

V Pardubicích dne 3. září 2018

## **PROHLÁŠENÍ**

Prohlašuji, že jsem tuto práci vypracoval/a samostatně. Veškeré literární prameny a informace, které jsem v práci využil/a, jsou uvedeny v seznamu použité literatury.

Byl/a jsem seznámen/a s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako Školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 30. 4. 2019

Bc. Jindřich Hegr

## **PODĚKOVÁNÍ:**

Tímto bych rád poděkoval svému vedoucímu práce doc. Jiřímu Křupkovi PhD. za jeho odbornou pomoc, cenné rady a poskytnuté materiály, které mi pomohly při zpracování diplomové práce. Také děkuji Fakultě elektrotechniky a informatiky Univerzity Pardubice, která mi umožnila pracovat v jedné ze svých laboratoří. Na této spolupráci se podílel především Ing. Daniel Honc, Ph.D. a cenné rady poskytli i tamější studenti, Bc. Martin Břeň a Ing. Dominik Štursa. V neposlední řadě také patří dík mé rodině a přítelkyni, kteří mi zajišťovali potřebné podmínky a morální oporu v celém průběhu vytváření této práce.

## **ANOTACE**

*Diplomová práce se zaměřuje na návrh nových systémů na bázi výpočetní inteligence pro řízení výukového modelu CE151 od firmy Humusoft. Před hlavními modifikacemi modelu se popíše samotný výukový model a identifikují se problémy s řízením. Původní způsob regulace se dále použije jako podklad pro učení výpočetní inteligence. Po získání trénovacího souboru dat se vyberou a definují vhodné metody výpočetní inteligence pro návrh nových regulátorů. Následně dojde k jejich implementaci, analýze jejich funkce a porovnání s původním regulátorem pomocí několika kritérií. Obsah práce může sloužit pro výzkumné účely v oblasti umělé a výpočetní inteligence v souvislosti v řízením.*

## **KLÍČOVÁ SLOVA**

*CE151, Humusoft, kulička na ploše, MATLAB, Simulink, neuronová síť, ANFIS, fuzzy řízení, regulátor*

## **TITLE**

*Model of control system on computational intelligence basis*

## **ANNOTATION**

*The diploma thesis focuses on design of new systems on computational intelligence (CI) basis for educational model CE151 by Humusoft Ltd. control. Before the main modifications, educational model alone will be depicted as well as issues with its controlling. Original way of control will be used as a foundation for computational intelligence learning. After the acquisition of training dataset, suitable methods of CI will be defined for the purpose of designing new controllers. After that, their implementation will be done as well as their analysis of functionality and comparing with original controller using few criterions. Contents of this thesis may be used for research purposes in the artificial or computational intelligence domains.*

## **KEYWORDS**

*CE151, Humusoft, ball & plate, MATLAB, Simulink, neural network, ANFIS, fuzzy controlling, controller*

# OBSAH

ÚVOD.....	11
<b>1 POPIS MODELU ŘÍZENÍ HUMUSOFT CE 151.....</b>	<b>12</b>
1.1 HW PRVKY MODELU .....	12
1.2 SW PRVKY MODELU .....	14
1.3 MATEMATICKÝ POPIS MODELU SE ZAMĚŘENÍM NA PID .....	15
1.4 PRŮBĚH PŮVODNÍ REGULACE .....	18
1.5 PROBLÉMY PŘI PRŮBĚHU REGULACE SYSTÉMU .....	19
1.6 OBECNÉ ÚPRAVY MODELU PRO LEPŠÍ REGULACI.....	21
1.6.1 Vylepšení snímání plošiny .....	21
1.6.2 Korekce řídicího systému plošiny .....	21
1.6.3 Zdokonalení kalibrace systému před regulací .....	22
<b>2 PŘÍPRAVA NA UČENÍ ŘÍDÍCÍCH SYSTÉMŮ NA BÁZI CI .....</b>	<b>24</b>
2.1 SYSTÉM PRO VYTVOŘENÍ SOUBORU DAT PRO UČENÍ CI .....	24
2.1.1 Logování dat z průběhu regulace.....	24
2.1.2 Průběh logovaných měření a export dat .....	25
2.2 KOMPLETACE SOUBORU DAT PRO UČENÍ CI.....	26
2.2.1 Načtení a předzpracování dat .....	27
2.2.2 Vytvoření samotných souborů.....	28
2.2.3 Charakteristiky zpracovávaných dat.....	29
2.3 OVĚŘENÍ KVALITY TRÉNOVACÍCH DAT .....	30
<b>3 NEURONOVÁ SÍŤ JAKO ŘÍDÍCÍ SYSTÉM.....</b>	<b>34</b>
3.1 TEORETICKÝ ÚVOD DO NEURONOVÝCH SÍTÍ .....	34
3.2 NÁVRH NEURONOVÉ SÍTĚ PRO REGULACI MODELU CE151 .....	36
3.3 IMPLEMENTACE NEURO-REGULÁTORU A JEHO ZHODNOCENÍ.....	40
<b>4 SPOJENÍ NEURONOVÉ SÍTĚ A FUZZY LOGIKY PRO ŘÍZENÍ SYSTÉMU.....</b>	<b>43</b>
4.1 TEORETICKÝ ÚVOD DO FUZZY MNOŽIN .....	43
4.2 TEORETICKÝ ÚVOD DO PROBLEMATIKY ANFIS.....	44
4.3 NÁVRH FIS VE SPOJENÍ S NEURONOVOU SÍŤÍ PRO REGULACI MODELU CE151 .....	45
4.4 IMPLEMENTACE REGULÁTORU NA BÁZI ANFIS A JEHO ZHODNOCENÍ.....	50
<b>5 POROVNÁNÍ NAVRŽENÝCH REGULÁTORŮ A JEJICH UMÍSTĚNÍ VE SCHÉMATU ŘÍDÍCÍHO SYSTÉMU .....</b>	<b>53</b>
5.1 NOVÉ SCHÉMA PRO ŘÍZENÍ MODELU V PROSTŘEDÍ MATLAB-SIMULINK .....	53
5.2 PRŮBĚHY REGULACE V ZÁVISLOSTI NA TYPU KULIČKY .....	54
5.2.1 Ping-pongový míček.....	54
5.2.2 Kovová kulička s pogumovaným povrchem .....	56
5.2.3 Kovová kulička.....	56
5.3 POROVNÁNÍ PRŮMĚRNÝCH ČASŮ STABILIZACE .....	57
5.4 NÁROČNOST NÁVRHŮ ŘÍDÍCÍCH SYSTÉMŮ.....	60
5.5 SLOŽITOST STRUKTURY REGULÁTORŮ.....	61
<b>ZÁVĚR.....</b>	<b>62</b>
<b>POUŽITÁ LITERATURA.....</b>	<b>63</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>65</b>

## SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1: Výukový model řízení Humusoft CE151 .....	12
Obrázek 2: Typy kuliček použitých pro řízení .....	13
Obrázek 3: Nezpracovaný obraz (raw picture) snímaný webkamerou modelu.....	13
Obrázek 4: Původní schéma pro řízení modelu CE151.....	14
Obrázek 5: Zpracovaný obraz (threshold) z webkamery modelu.....	15
Obrázek 6: Blokové schéma číslicového PID regulátoru.....	17
Obrázek 7: Detailní schéma původního PID regulátoru.....	18
Obrázek 8: Přechodová charakteristika typického průběhu regulace PID regulátorem.....	19
Obrázek 9: Trajektorie kuličky pro typický průběh regulace původním PID regulátorem.....	19
Obrázek 10: Problém s velkým nakláněním plošiny .....	20
Obrázek 11: Hodnoty akční veličiny regulátoru po stabilizaci soustavy - původní kalibrace .	20
Obrázek 12: Nehybná kulička při výrazném náklonu plošiny .....	21
Obrázek 13: Parametry nově nastaveného vstupu do soustavy.....	22
Obrázek 14: Zdokonalená kalibrace systému před začátkem regulace .....	23
Obrázek 15: Hodnoty akční veličiny regulátoru po stabilizaci soustavy - nová kalibrace.....	23
Obrázek 16: Schéma systému pro logování dat z průběhu regulace .....	24
Obrázek 17: Trajektorie kuličky při logovaném měření s manuální destabilizací systému.....	25
Obrázek 18: Příkaz pro export dat z MATLABu do souboru CSV .....	26
Obrázek 19: Schéma v IBM SPSS Modeler pro kompletaci trénovacích dat CI .....	27
Obrázek 20: Načtení jednotlivých logů provedených měření .....	27
Obrázek 21: Načtení a předzpracování dat jednotlivých logů v IBM SPSS Modeler.....	28
Obrázek 22: Vytvoření souborů dat pro učení CI v prostředí IBM SPSS Modeler .....	29
Obrázek 23: Charakteristiky datového souboru pro učení CI .....	30
Obrázek 24: Evaluační graf ověřování dat pro učení CI pomocí neuronové sítě.....	32
Obrázek 25: Nejlepší ověřovací model neuronové sítě pro dva výstupy .....	32
Obrázek 26: Míra významu prediktorů ověřovací neuronové sítě .....	33
Obrázek 27: Jednoduchý model neuronu .....	35
Obrázek 28: Import souboru dat pro učení neuronové sítě .....	36
Obrázek 29: Vlastnosti použitých souborů dat pro učení neuronové sítě .....	37
Obrázek 30: Výsledky učení neuronové sítě .....	37
Obrázek 31: Histogram chyb po učení neuronové sítě.....	38
Obrázek 32: Detaily průběhu učení neuronové sítě.....	39
Obrázek 33: Průběh učení neuronové sítě v grafu.....	39
Obrázek 34: Schéma struktury neuronové sítě pro řízení modelu.....	40
Obrázek 35: Detailní schéma navrženého neuro-regulátoru v prostředí MATLAB-Simulink	41
Obrázek 36: Příklad pro funkci příslušnosti fuzzy množiny "mladý člověk" .....	43
Obrázek 37: Architektura systému ANFIS pro dvou vstupový FIS typu Sugeno.....	45
Obrázek 38: Nastavení parametrů pro vygenerovaný FIS.....	46
Obrázek 39: Nastavení parametrů pro trénování systému ANFIS .....	47
Obrázek 40: Testování systému ANFIS .....	48
Obrázek 41: Struktura vygenerovaného systému ANFIS .....	48
Obrázek 42: Funkce příslušnosti použité v systému ANFIS.....	49
Obrázek 43: Struktura neuronové sítě v systému ANFIS .....	50
Obrázek 44: Detailní schéma ANFIS-regulátoru .....	51
Obrázek 45: Modifikované schéma pro řízení modelu CE151 .....	54
Obrázek 46: Přechodová charakteristika typického průběhu regulace regulátory CI .....	55
Obrázek 47: Trajektorie kuličky pro typický průběh regulace ANFIS a neuro-regulátorem...	56
Obrázek 48: Charakteristiky dob regulace ping-pongového míčku .....	58
Obrázek 49: Charakteristiky dob regulace kovové kuličky s pogumovaným povrchem .....	59
Obrázek 50: Charakteristiky dob regulace kovové kuličky.....	59



Tabulka 1: Doplnující informace o zpracovávaných datech .....	30
Tabulka 2: Přesnost predikce ověřovací NS v závislosti na různých vstupech.....	31
Tabulka 3: Složky soft computingu.....	34
Tabulka 4: Průměrné časy stabilizace RO pomocí neuro-regulátoru .....	42
Tabulka 5: Průměrné časy stabilizace RO pomocí ANFIS-regulátoru.....	52
Tabulka 6: Srovnání průměrných směrodatných odchylek všech regulačních pochodů.....	60

## SEZNAM ZKRATEK A SYMBOLŮ

ANFIS	Adaptivní neuro-fuzzy inferenční systém
AI	Umělá inteligence (Artificial Intelligence)
CI	Výpočetní inteligence (Computational Intelligence)
CE151	Označení výukového modelu pro řízení
CSV	Souborový formát (Comma-separated values)
$e$	Regulační odchylka regulátoru
HW	Hardware
IBM	Společnost International Business Machines
MIMO	Systém s několika vstupy a několika výstupy
MISO	Systém s několika vstupy a jedním výstupem
NS	Neuronová síť
PID	Regulátor, složený z proporcionální, integrační a derivační části
$q_i$	Stavitelné parametry regulátoru
$R$	Korelační koeficient
RO	Regulační obvod
s	Sekunda
SPSS	Firma Statistical Package for the Social Sciences
SW	Software
$T_s$	Vzorkovací perioda regulačního systému
$u$	Akční veličina regulátoru

# ÚVOD

Cílem této diplomové práce je navrhnout a implementovat model řízení na bázi výpočetní inteligence pro výukový model CE151 od firmy Humusoft. Nejprve se práce zaměří na popis samotného výukového modelu a původního způsobu řízení. Dále se definují metody výpočetní inteligence, kterých se využije pro návrh nových systémů pro řízení CE151. Tyto metody se poté aplikují v návrhu nových způsobů regulace. Posledním krokem bude implementace systémů výpočetní inteligence jako regulátorů pro model CE151 a jejich porovnání.

Při popisu modelu se práce zaměří nejen na hardwarové a softwarové prvky, ale také na samotný průběh regulačního pochodu. V rámci příprav na učení výpočetní inteligence bude snaha model upravit tak, aby se minimalizovala šance vzniku neočekávaných chyb a v očích CI iracionálního chování původního regulátoru.

Po identifikaci a zmírnění všech nežádoucích vlivů při procesu regulace se bude práce věnovat získání a úpravě dat, kterých se využije při učení výpočetní inteligence. V této části se navrhne systém pro získání dat, která charakterizují jednotlivé regulační pochody. Poté se tato data zkompletují a proběhne jejich úprava, která se bude podřizovat nástrojům pro návrh systémů CI. Ještě před použitím datových souborů se data otestují a proběhne prvotní návrh struktury nových regulátorů.

Následně se práce zaměří na popis vybraných metod CI a jejich využití pro výukový model CE151. Po teoretickém úvodu se představí a charakterizují nástroje pro návrh systémů CI a jejich praktické využití v průběhu návrhu nových způsobů řízení. Navržené systémy CI se následně detailně popíší, především jejich přesnost.

Nově navržené systémy CI se poté implementují do regulačního systému výukového modelu jako regulátory na bázi výpočetní inteligence. Následně proběhne otestování jejich funkce a stručná charakteristika regulačního pochodu.

Na závěr práce zhodnotí všechny typy regulátorů. Hlavními kritérii při porovnávání bude doba regulace a míra adaptability se zaměřením na rozdíly mezi původním způsobem regulace a regulace na bázi CI. Zhodnotí se také složitost struktury jednotlivých regulátorů, s tím spojené nároky na výpočetní výkon a náročnost jejich návrhu.

# 1 POPIS MODELU ŘÍZENÍ HUMUSOFT CE 151

V této části se práce zaměří na popis modelu, který je předmětem regulace. Z pohledu celého systému se jedná o soustavu, která je řízena pomocí počítače. Přesněji je regulátor implementován pomocí simulace v prostředí programu MATLAB, modulem Simulink. Hardwarové a softwarové části systému se práce bude věnovat pouze stručně. Blíže se tímto popisem zabývá [3], kde jsou detailně charakterizovány i všechny parametry částí modelu. Více se popis zaměří na původní PID regulátor a nově nalezené problémy s procesem stabilizace ve vztahu na pozdější návrh CI.

## 1.1 HW prvky modelu

Hlavním hardwarovým prvkem modelu je řízená soustava, vyobrazená na fotografii níže (viz Obrázek 1). Obecně lze rozdělit do několika částí. Náklonnou plošinou se řídí pozice kuličky. Tuto plošinu ovládají pomocí kovových lanek krokové motory umístěné v boxu pod ní. Zmiňované části jsou včetně držáku na webkameru kovové. Celá soustava je napájena síťovým proudem pomocí externího zdroje.



**Obrázek 1:** Výukový model řízení Humusoft CE151

*Zdroj: vlastní*

Dále se do soustavy řadí samotné kuličky (viz Obrázek 2), které se řídicí systém snaží vyrovnat na předem stanovenou pozici na plošině. V práci je nejčastěji využíván běžný, ping-pongový míček. Pro ověření adaptability systému jsou však použity i kovové kuličky jiné velikosti, z nichž jedna má pogumovaný povrch.



**Obrázek 2:** Typy kuliček použitých pro řízení

*Zdroj: vlastní*

Nad plošinou je na držáku umístěna webkamera, která snímá pozici kuličky na plošině. Tato webkamera je základem pro zpětnou vazbu celého systému. Perioda zaznamenávání obrazu je nastavena na 1 snímek za 0,15s. Pro řídicí systém je důležité, aby byla webkamera upevněna nad plošinu v takové vzdálenosti, aby snímala pouze ji. V opačném případě by mohl systém vyhodnocovat obraz mimo desku jako pozici kuličky, nebo kuličku „nevidět“ vůbec. Příklad snímaného obrazu ještě před zpracováním je znázorněn na obrázku níže (viz Obrázek 3).



**Obrázek 3:** Nezpracovaný obraz (raw picture) snímaný webkamerou modelu

*Zdroj: vlastní*

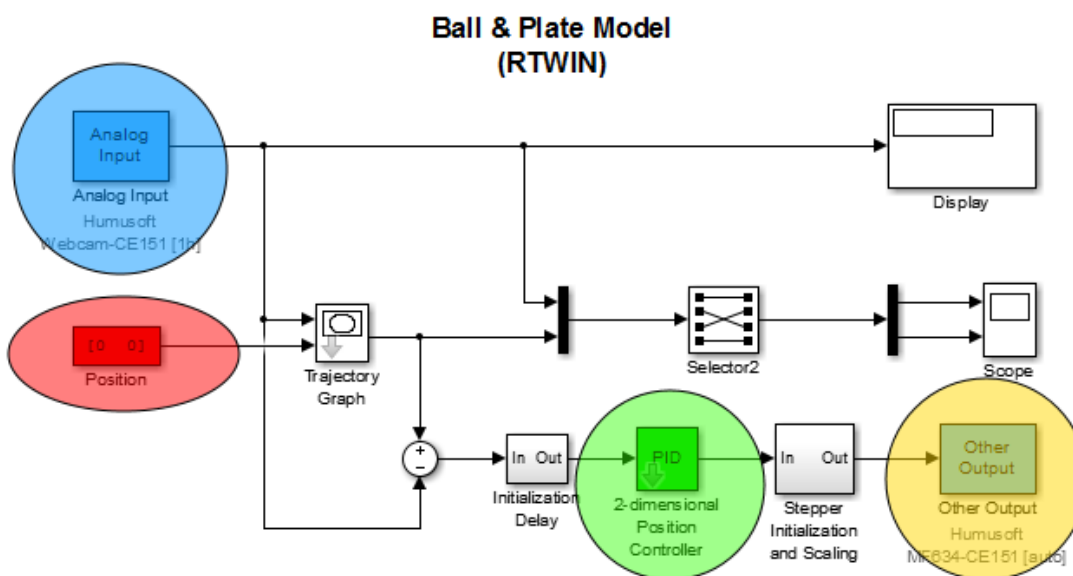
Hardwarovou částí modelu CE151 je také samozřejmě i počítač, na kterém je softwarově řešen regulační proces systému. Spadá sem i veškerá kabeláž pro napájení všech elektronických komponent systému a přenos řídicích signálů mezi regulátorem a soustavou.

## 1.2 SW prvky modelu

Softwarová část modelu se stará o regulační proces systému. V podstatě se jedná o simulaci regulátoru pomocí modulu Simulink v prostředí MATLAB. MATLAB je výkonné interaktivní prostředí pro vědecké výpočty. Do jednoho prostředí spojuje technické výpočty, vizualizaci dat a programovací jazyk. Simulink je jedním z neznámějších a nepoužívanějších rozšíření MATLABu. Využívá MATLAB pro modelování, simulaci a analýzu dynamických systémů v přehledném grafickém uživatelském prostředí. Název tohoto rozšíření je zkratka dvou slov z angličtiny – SIMULATION a LINK. Aplikaci vyvíjí firma MathWorks. [7]

Nejen samotný MATLAB, ale i Simulink má své rozšiřující komponenty, nazývané toolboxy. Pro původní model řízení modelu nebylo nutné žádný doplňkový toolbox instalovat, samotný Simulink obsahoval všechny potřebné funkce. Při návrhu regulátorů na bázi výpočetní inteligence je však bude nutné využít. Jmenovitě jde o Fuzzy Logic Toolbox, který rozšiřuje prostředí o práci s fuzzy logikou a Neural Network Toolbox, díky čemuž se dá snadno zkonstruovat neuronová síť.

Schéma řídicího systému modelu CE151 v prostředí Simulink (viz Obrázek 4), ze kterého práce vychází se může rozdělit do několika základních částí.

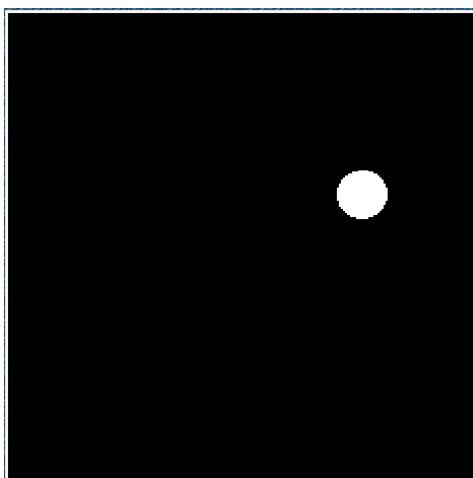


**Obrázek 4:** Původní schéma pro řízení modelu CE151

*Zdroj: upraveno dle [4]*

První, modře označená část schématu symbolizuje vstup do regulátoru. V tomto bloku dochází ke zpracování prvotních („raw“) dat, které se získávají z webkamery nad náklonnou plošinou. Systém se v této fázi snaží zjistit polohu kuličky, přesněji její relativní souřadnice od středu snímané plochy. Protože se v tomto kroku preferuje rychlost, nesnaží se systém náročně

vyhodnotit snímaný obraz, ale provede tzv. prahování (thresholding). Tím obraz převede na černobílý snímek (viz Obrázek 5), kde poměrně snadno určí těžiště nalezeného tvaru kuličky.



**Obrázek 5:** Zpracovaný obraz (threshold) z webkamery modelu

*Zdroj: vlastní*

V červeně označené části schématu uživatel nastavuje cíl regulace. Přesněji souřadnice  $x$  a  $y$ , na které se má kulička na plošině stabilizovat.

Samotná regulace se provádí v bloku „PID“, který je ve schématu výše označen zelenou barvou. Tímto blokem se bude blíže zabývat následující kapitola (viz 1.3).

Nakonec v systému probíhá zesílení signálu akční veličiny (tedy výstupu regulátoru) a modře označenou částí schématu systém vysílá pokyny pro krokové motory, ovládající náklon plošiny. Systém se řídí v reálném čase a jak již bylo zmíněno, celý tento proces se vykonává jednou za 0,15 s.

### 1.3 Matematický popis modelu se zaměřením na PID

Dle [1] můžeme regulační obvod rozdělit na dvě hlavní části. Je jimi soustava a regulátor, který soustavu řídí. Tyto části se dají matematicky popsat pomocí rovnic přenosu.

Dle manuálu má přenos soustavy tvar [4]:

$$F(s) = \frac{4.803}{s^2(0.187s+1)} \quad (1)$$

Jak je patrné z charakteristické rovnice, tedy jmenovatele výše zmíněného přenosu, soustava vykazuje astatismus druhého řádu. Taková soustava nemá samoregulační člen. To znamená, že po změně signálu na vstupu se systém destabilizuje a s rostoucím časem se tato destabilizace zvyšuje. V [1] je astatismus definován jako setrvačnost a celá soustava se nazývá integrační. Charakteristiky astatismu jsou v případě modelu CE151 snadno viditelné. Pokud bychom

ze systému odebrali regulátor a pouze naklonili plošinu, kulička podle ní sjede vlivem gravitace až na její kraj, který symbolizuje maximální povolené vychýlení soustavy. Protože stabilita systému je závislá na poloze kuličky, došlo by tedy při změně vstupu (náklonu plošiny) k postupné destabilizaci až na maximum, kterého je soustava schopna.

Regulátor je matematicky popsán rovnicí přenosu [8]:

$$R(s) = 0,34 + \frac{0,23}{s} + 0,26s \quad (2)$$

Regulátor použitý pro model CE151 je typu PID. Jak jeho označení napovídá, tento typ se skládá ze tří částí: proporcionální, integrační a derivační. V uzavřeném regulačním obvodu složka I odstraňuje trvalou regulační odchylku a vlivem složky D zlepšuje vlastnosti stability regulačního obvodu. V počátku přechodového děje převládá derivační složka regulátoru, v narůstajícím čase převládá integrační složka regulátoru. [1]

Pro řízení v diskretních regulačních obvodech se velmi často používají diskretní algoritmy regulace vycházející z odpovídajících konvenčních spojitých (analogových) algoritmů. Činnost ideálního spojitého PID regulátoru lze popsat rovnicí (3)

$$u(t) = k_R \left[ e(t) + \frac{1}{T_I} \int e(\tau) d\tau + T_D \frac{de(t)}{dt} \right], \quad (3)$$

kde:  $k_R$  představuje zesílení regulátoru,  $T_I$  je integrační časová konstanta,  $T_D$  je derivační časová konstanta. [1]

Řízení však u modelu CE151 probíhá diskretně, s časovou konstantou vzorkování 0,15 s. Proto je nutné rovnici upravit. Protože se zde neobjevuje spojitý signál v každém okamžiku, je vhodné použít diferenční rovnici, díky které zjistíme hodnotu akční veličiny v každém vzorkovacím okamžiku [1]:

$$u(kT) = k_R e(kT) + \frac{k_R}{T_I} I(kT) + k_R T_D D(kT). \quad (4)$$

Tato rovnice předpokládá znalost hodnoty derivace  $D(kT)$  a integrálu  $I(kT)$  v daném diskretním časovém okamžiku  $k$ . V praxi jsou však tyto matematické operace nahrazeny jednoduššími. Pokud nahradíme integraci zpětnou obdélníkovou metodou (také stupňovitá náhrada zpět) a derivaci zpětnou diferencí, dostaneme univerzálně platnou diferenční rovnici přírůstkového algoritmu ve tvaru [1]:

$$u(kT) = q_0 e(kT) + q_1 e[(k-1)T] + q_2 e[(k-2)T] + u[(k-1)T], \quad (5)$$

kde tzv. stavitelné parametry  $q_0$ ,  $q_1$  a  $q_2$  jsou funkcí  $k_R$ ,  $T_I$ ,  $T_D$ , periody vzorkování  $T$  a metody náhrady integrálu, tedy platí [1]:



$$q_0, q_1, q_2 = f(k_R, T_I, T_D, T). \quad (6)$$

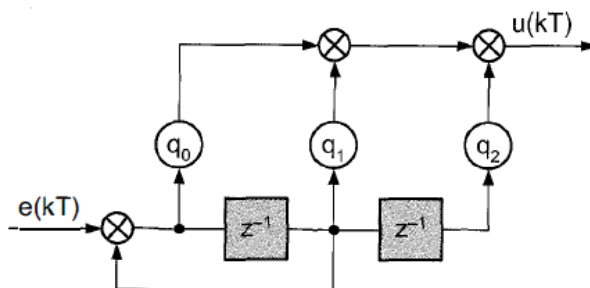
Z-transformací rovnice (5) dostaneme přenos diskrétního PID regulátoru. Tento tvar rovnice je vhodný pro modelování nových systémů řízení, kterým se práce věnuje v pozdějších kapitolách [1]:

$$G_R(z) = \frac{U(z^{-1})}{E(z^{-1})} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}}. \quad (7)$$

Tento přenos se může schematicky zakreslit, jak je znázorněno na obrázku níže (viz Obrázek 6). Z tohoto schématu a univerzálně platné diferenční rovnice přírůstkového algoritmu (5) jasně vyplývá, že diskrétní (číslíkový) PID regulátor řídí soustavu, přesněji akční veličinu  $u(kT)$ , díky čtyřem veličinám. Jsou jimi:

- regulační odchylka v daném vzorkovacím okamžiku  $T$ :  $e(kT)$ ,
- regulační odchylka zpožděná o jeden  $T$ :  $e[(k-1)T]$ ,
- regulační odchylka zpožděná o dva  $T$ :  $e[(k-2)T]$ ,
- akční veličina zpožděná o jeden  $T$ :  $u[(k-1)T]$ .

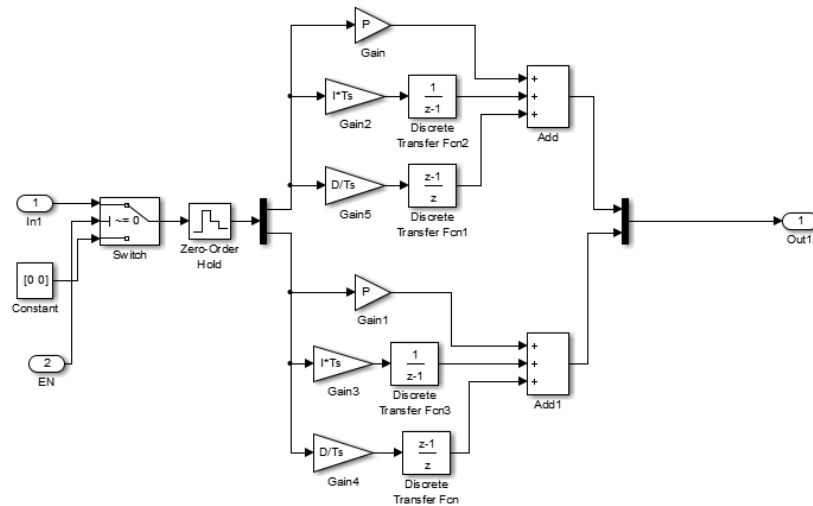
Toto je naprosto klíčová informace pro návrh regulátorů na bázi CI pro model CE151, kdy výpočetní inteligence nahradí tradiční PID regulátor.



**Obrázek 6: Blokové schéma číslicového PID regulátoru**

*Zdroj: [1]*

Z detailnějšího zkoumání procesu regulace modelu CE151 vyplývá, že model pro řízení využívá dva, paralelně operující, PID regulátory. Znamená to tedy, že každá osa náklonu plošiny se řídí nezávisle na sobě. Tento regulační proces je znázorněn na schématu níže (viz Obrázek 7, vyšší rozlišení schématu uvedeno v příloze C). Vstupní signál se nejprve drží na hodnotách  $[0 \ 0]$ , dokud se nedokončí kalibrace. Poté se rozdělí do dvou složek, přičemž každou z nich zpracovávají bloky pro zesílení (proporcionální složka), integraci a derivaci. Výstupy z těchto bloků jsou sečteny a tento součet reprezentuje akční veličinu pro každou z os náklonu plošiny.



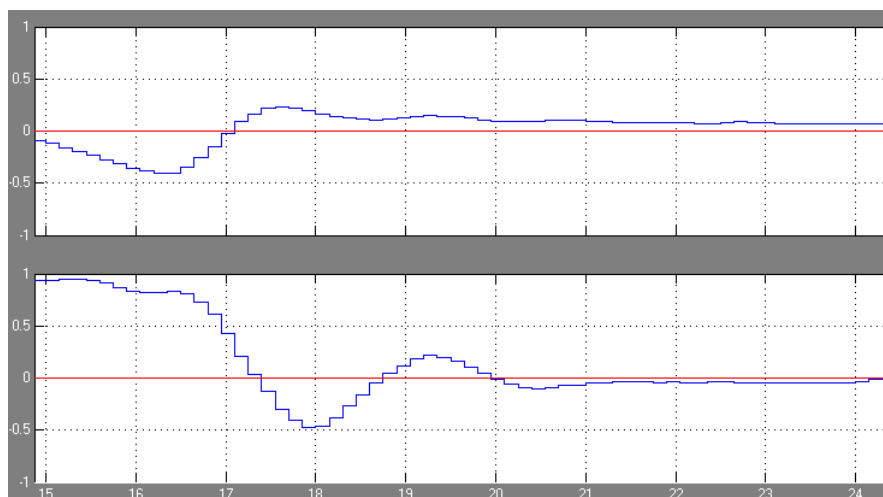
**Obrázek 7:** Detailní schéma původního PID regulátoru

*Zdroj: převzato z mdl-souboru k [4]*

PID regulátor se optimalizuje pro danou soustavu pomocí různých metod. V tomto procesu se počítají a nastavují tři parametry: P, I a D a sleduje se, zda takto nastavený PID regulátor stabilizuje soustavu, případně po jakém čase. Velikost těchto parametrů značí zapojení každé ze složek regulátoru do řídicího procesu. V případě modelu CE151 je nastavení následující: složka P=0,6, složka I=0,1 a složka D=0,3.

## 1.4 Průběh původní regulace

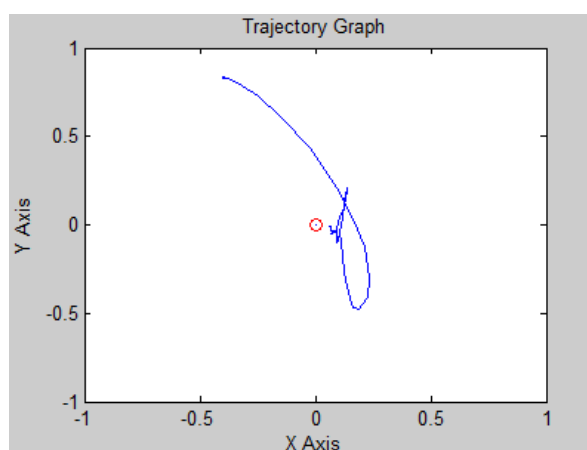
Průběh regulace můžeme znázornit pomocí přechodové charakteristiky. Typickou přechodovou charakteristiku modelu CE151 vizualizuje graf níže (viz Obrázek 8). Osa  $x$  představuje čas regulace, osa  $y$  poté výstup systému. Červená osa představuje cíl regulace v čase. Modrá osa představuje stav systému v daném čase, přesněji polohu kuličky na náklonné plošině. Porovnáním obou os tedy dostáváme informaci o tom, jak probíhal proces regulace. Přechodová charakteristika modelu CE151 má 2 složky, protože dochází ke stabilizaci ve dvou osách, a to nezávisle na sobě. Proto jsou na obrázku níže vyobrazeny 2 grafy.



**Obrázek 8:** Přechodová charakteristika typického průběhu regulace PID regulátorem

*Zdroj: vlastní*

Po několika měřeních se ukázalo, že hodnota regulační odchylky v ustáleném stavu, kdy se systém stabilizoval, byla  $\pm 5\%$  od cíle regulace. Toto je ovlivněno zejména nedokonalostmi povrchu ping-pongového míčku, nicméně tím se bude práce zabývat v následující kapitole. Tuto přechodovou charakteristiku pro lepší představu o pohybu kuličky znázorňuje obrázek níže (viz Obrázek 9). Jak je zřejmé, v ose  $y$  se soustava v tomto případě stabilizovala s téměř nulovou regulační odchylkou. V ose  $x$  je ale odchylka větší. V grafu je regulační odchylka viditelná v porovnání s cílem regulace, na grafu trajektorie tuto odchylku představuje rozdíl mezi koncem modré křivky a červené kružnice, představující cíl regulace. Dle [13] se tento typ regulačního procesu nazývá kmitavý pochod s překmity (přeregulováním).



**Obrázek 9:** Trajektorie kuličky pro typický průběh regulace původním PID regulátorem

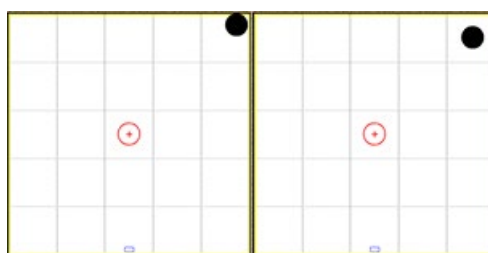
*Zdroj: vlastní*

## 1.5 Problémy při průběhu regulace systému

Při návrhu regulátorů v rámci simulací se většinou nezabýváme poruchovou veličinou, protože má velmi náhodný charakter a je složité ji matematicky popsat. Pro případ modelu

CE151 je však vhodné tyto poruchy alespoň identifikovat a vzít je, pokud možno, v potaz při návrhu, a hlavně hodnocení nových regulátorů.

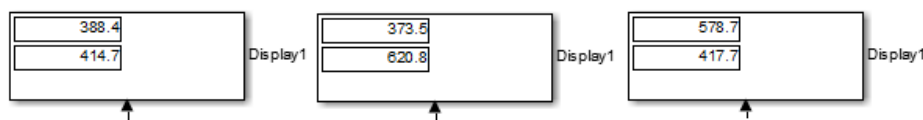
Jak bylo zmíněno v kapitole 1.3, nové regulátory se budou pravděpodobně ovládat na základě několika veličin, které budou opožděné. Tyto předchozí stavy mají velký dopad na regulaci, proto prvním identifikovaným problémem je maximální náklon plošiny. Tento náklon totiž způsobí oddálení kuličky od kamery tím dojem, že je kulička blíže středu plošiny. Na obrázku níže (viz Obrázek 10) je tento problém vystižen. V levé části je plošina vodorovná, v pravé došlo k maximálnímu vychýlení pravého, horního rohu plošiny. Nový řídicí systém tedy může dojít k závěru, že čím více se plošina bude vychylovat, tím blíže bude soustava stabilizaci. Toto však není pravda a může to vést nejen k destabilizaci, ale i zničení celého modelu.



**Obrázek 10:** Problém s velkým nakláněním plošiny

*Zdroj: vlastní*

Dalším problémem řízení, který se může zařadit do poruchové veličiny je rozdílnost výstupu regulátoru v ustáleném stavu systému. Tato skutečnost je znázorněna na obrázku níže (viz Obrázek 11), kdy došlo ke třem nezávislým měřením stabilizace modelu. Výsledné výstupy regulátoru, tedy hodnota akční veličiny, se ve výsledku lišila v řádech stovek. Toto pro PID regulátor sice není problémem, nicméně může to mít za následek složitější proces učení výpočetní inteligence a následně i složitější způsob regulace pomocí CI. Tento problém je pravděpodobně zapříčiněn prokluzováním lanek, které řídí náklon plošiny.



**Obrázek 11:** Hodnoty akční veličiny regulátoru po stabilizaci soustavy - původní kalibrace

*Zdroj: vlastní*

Posledním závažnějším problémem jsou při regulaci nedokonalosti na povrchu kuličky. Ty zapříčiňují, že kulička nezmění svoji polohu i přes náklon plošiny. Toto se stává hlavně při menších náklonech, kterých regulátor využívá při menší regulační odchylce. Kulička by tak teoreticky neměla nabrat takovou setrvačnost a posunout se jen o malý kousek k cíli regulace. Místo toho se však kulička zastaví o hranu na svém povrchu a plošina se pomalu naklání vlivem

složky I. V určitém okamžiku se kulička dostane přes tuto hranu, nicméně to už je plošina vychýlená příliš a dojde ke značné destabilizaci.



**Obrázek 12:** Nehybná kulička při výrazném náklonu plošiny

*Zdroj: vlastní*

Pokud se výše popsáný problém projeví několikrát při regulaci, značně to zkresluje dobu regulace a následné vyhodnocování efektivity řízení. Protože se v takové chvíli chová systém iracionálně, mohlo by také dojít při návrhu CI ke značnému zkomplikování procesu učení a následně i ke zbytečné složitosti regulace.

## **1.6 Obecné úpravy modelu pro lepší regulaci**

Aby probíhal návrh nových řídicích systému na bázi výpočetní inteligence co nejlépe a bez zbytečných chyb, navrhly se v rámci této práce obecné úpravy regulačního systému, které částečně řeší problémy, které byly popsány v předchozí kapitole.

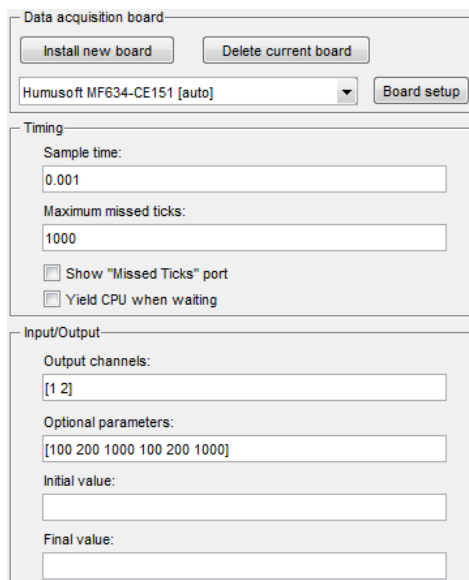
### **1.6.1 Vylepšení snímání plošiny**

Aby se zamezilo problému ohledně přibližování kuličky se středu při maximálním naklonění plošiny, upravil se způsob jejího snímání. Přesněji se k ní přiblížila webkamera tak, aby ve vodorovné poloze plošiny snímala pouze polovinu kuličky. Systém pro vyhodnocování obrazu dokáže poměrně přesně určit těžiště i tvarů, které nejsou oválné. Při maximálním vychýlení plošiny tedy webkamera „vidí“ kuličku celou a při vodorovné plošině pouze její půlku. Ve výsledku se docílilo toho, že je toto zkreslení polohy vlivem náklonu plošiny pouze minimální.

### **1.6.2 Korekce řídicího systému plošiny**

Rozdílná akční veličina při ustáleném stavu regulace se částečně vyřešila nastavením menší vzorkovací frekvence výstupního bloku regulačního systému (viz Obrázek 4). Jak napovídá obrázek níže (viz Obrázek 13) se všemi nastavitelnými parametry, vzorkovací perioda (sampling time) se z původních 0,01 s zmenšil desetkrát, tedy na 0,001 s. S tím je spojen

i parametr maximálního počtu vynechaných taktů. Ten se musel zvýšit na 1000. I když zbytek bloků regulačního systému pracuje s periodou 0,15 s, docílilo se mnohem plynulejšího vychylování plošiny. Tento způsob náklonu plošiny povede ke snížení prokluzování lanek soustavy.



**Obrázek 13:** Parametry nově nastaveného vstupu do soustavy

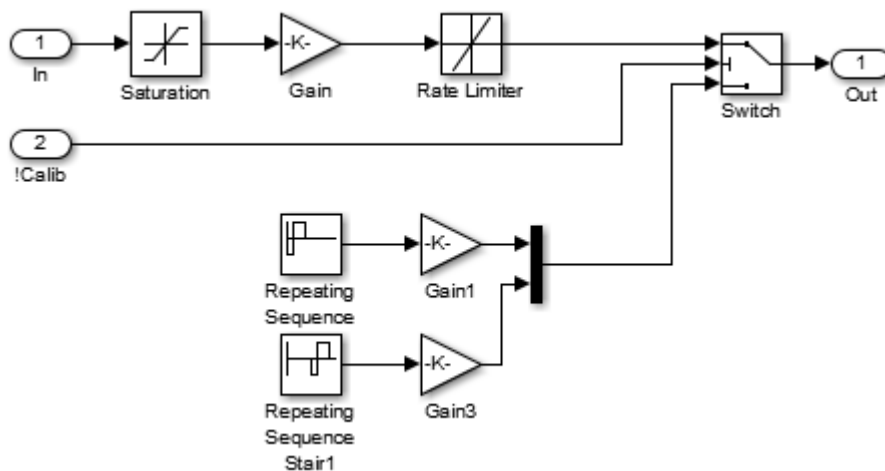
*Zdroj: vlastní*

### 1.6.3 Zdokonalení kalibrace systému před regulací

Vždy před spuštěním regulace model CE151 provede kalibraci. Tím by se mělo docílit toho, že se určí hodnoty akčního členu, které odpovídají vodorovné poloze plošiny, a to nezávisle na náklonu plošiny před začátkem kalibrace. Problém však spočíval v tom, že při jistých náklonech plošiny před začátkem kalibrace se plošina vychýlila v krátkém čase až příliš, a to způsobilo proklouznutí lanek řídicích plošinu. Toto proklouznutí bylo vždy náhodné, proto se akční veličina po stejných procesech stabilizace ustálila na jiných hodnotách. Proto se systém kalibrace upravil.

Detailní schéma vylepšené kalibrace je znázorněno na obrázku níže (viz Obrázek 14). Její umístění v modifikovaném obecném schématu systému pro regulaci v prostředí MATLAB-Simulink je označeno zelenou barvou v kapitole 5.1 (viz Obrázek 45). Aby nedocházelo k neočekávaným situacím, řídí začátek regulace nově jednotkový skok zvláště pro všechny regulátory a kalibraci. Jednotkový skok změní stav vždy po 16 s po spuštění simulace, protože to je čas zmiňované kalibrace před regulací. Tento signál vypne umělé generování nulové regulační odchylky na vstupu regulátorů a vypne také kalibrování plošiny. Nový systém kalibrace vychyluje plošinu pomaleji a ve všech směrech. Tím nedochází k prokluzování lanek.

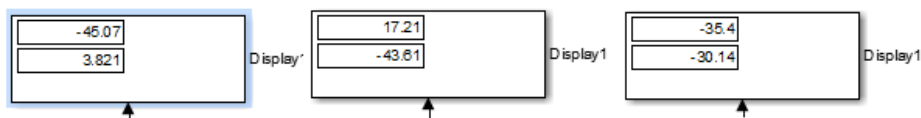
Dále definuje vodorovný stav plošiny při nulovém výstupu regulátoru (akční veličině) a to jak pro náklon v ose  $x$ , tak v ose  $y$ .



**Obrázek 14:** Zdokonalená kalibrace systému před začátkem regulace

*Zdroj: vlastní*

Zlepšená kalibrace systému a plynulejší náklon plošiny měly za následek minimální prokluzování lanek, které naklání plošinu. To dokázaly 3 nezávislé pokusy, kdy se měřila akční veličina vždy po dokončení stabilizace soustavy (viz Obrázek 15). Výstupy se liší v řádech desítek oproti stovkám při původním způsobu řízení (viz Obrázek 11).



**Obrázek 15:** Hodnoty akční veličiny regulátoru po stabilizaci soustavy - nová kalibrace

*Zdroj: vlastní*

## 2 PŘÍPRAVA NA UČENÍ ŘÍDICÍCH SYSTÉMŮ NA BÁZI CI

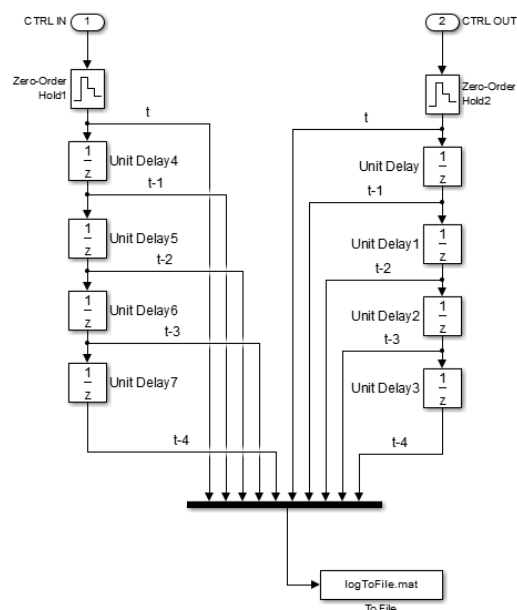
V této části se bude práce zabývat návrhem systému pro tvorbu souboru dat, který se následně použije pro učení regulátorů na bázi výpočetní inteligence. Prvním krokem je získání dat z několika průběhů regulace, dále se tato data musí upravit a sloučit do jednoho souboru. Po kompletaci se ještě ověří, zda jsou tato data vhodná pro učení CI.

### 2.1 Systém pro vytvoření souboru dat pro učení CI

Tato práce se zabývá výpočetní inteligencí, která napodobuje původní systém řízení. Při návrhu je tedy zapotřebí CI naučit pomocí dat, která vystihují dřívější regulační pochody, tedy procesy regulace. Protože je celá regulace simulována v prostředí Simulink, je relativně snadné vytvořit systém, který bude tato data zaznamenávat, tedy logovat.

#### 2.1.1 Logování dat z průběhu regulace

Logování je proces, kdy dochází k ukládání zvolených dat pro další analýzy. V případě modelu CE151 se budou logovat data ze vstupu (regulační odchylky) a výstupu (akční veličiny) původního PID regulátoru. V kapitole 1.3 je popsáno, které veličiny přesně mají vliv na měření, nicméně pro ověření a případné zdokonalení se budou ukládat 4 předchozí stavy jak vstupu, tak výstupu. Detailní schéma systému pro logování dat z regulačního pochodu je znázorněn na obrázku níže (viz Obrázek 16). V obecném modifikovaném schématu je logování označeno světle modrou barvou (viz Obrázek 45).



Obrázek 16: Schéma systému pro logování dat z průběhu regulace

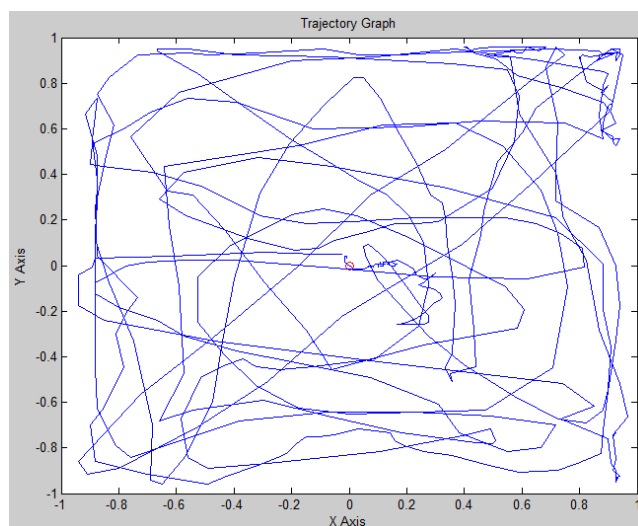
Zdroj: vlastní



V první fázi se jednotlivé signály ze vstupu a výstupu PID rozdělí na 5 složek. 4 složky se postupně zpozdí o jednu vzorkovací periodu, tedy 0,15 s. Už v průběhu logování se tedy získají hodnoty regulačního pochodu zpožděné o vzorkovací periody  $T_s$ ,  $2T_s$ ,  $3T_s$  a  $4T_s$ . Poté se všechny tyto složky sloučí a uloží do pracovního paměti prostředí MATLAB. Přesněji se tedy jedná o data vstupů a výstupů regulátoru v časech  $k$ ,  $k-1$ ,  $k-2$ ,  $k-3$  a  $k-4$  a to pro každou osu řízení náklonu plošiny. Takový soubor dat tedy bude obsahovat 20 atributů. Blok „Zero hold order“ zajistí, aby se v průběhu jednoho  $T_s$  udržovala v systému logování jedna hodnota a nedocházelo k neočekávaným chybám. Zpoždování signálu se provádí diskrétně, nepoužívá se zpoždění o čas, ale vzorkovací periodu.

### 2.1.2 Průběh logovaných měření a export dat

Pro tvorbu trénovacího souboru dat se logoval průběh dvaceti regulačních pochodů. Využilo se všech obecných úprav pro zlepšení regulace, popisovaných v kapitole 1.6. U dvou z těchto měření se soustava přibližně 3 minuty manuálně destabilizovala, aby se zaznamenalo co nejvíce dat při velkých regulačních odchylkách. Trajektorii kuličky při tomto logovaném měření můžeme vidět na obrázku níže (viz Obrázek 17). V praxi takové měření vypadalo tak, že se kulička připevnila na tmavý objekt, kterým se poté pohybovalo tak, aby se kulička dostala do co nejvíce souřadnic na plošině, hlavně tedy těch extrémních na okraji plošiny.

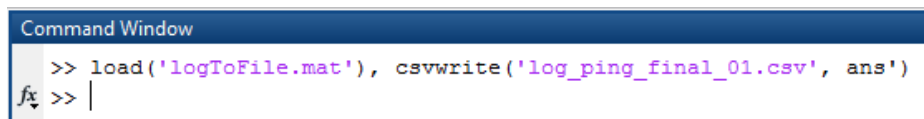


**Obrázek 17:** Trajektorie kuličky při logovaném měření s manuální destabilizací systému

*Zdroj: vlastní*

Po každém logovaném regulačním pochodu bylo nutné samotný soubor vyexportovat z pracovní paměti MATLABu do snáze zpracovatelné formy. Zpracování a posouzení dat by bylo samozřejmě možné i v prostředí MATLAB, nicméně pro srozumitelnější popis zpracovávání dat se zvolil jiný nástroj. Export se u každého měření provedl pomocí příkazového

řádku hlavního okna MATLABu (viz Obrázek 18). Pomocí příkazu „load“ se nahrála data z provedeného měření do proměnné pracovní paměti. Poté následoval příkazem „csvwrite“ export dat do CSV souboru, uloženého do právě otevřeného adresáře. „Csvwrite“ má dva argumenty: název exportovaného souboru a zdroj dat. V tomto případě se jako zdroj použila proměnná „ans“ s parametrem pro transponování matice, tedy uvozovkou za názvem proměnné.



```
Command Window
>> load('logToFile.mat'), csvwrite('log_ping_final_01.csv', ans')
fx >> |
```

**Obrázek 18:** Příkaz pro export dat z MATLABu do souboru CSV

*Zdroj: vlastní*

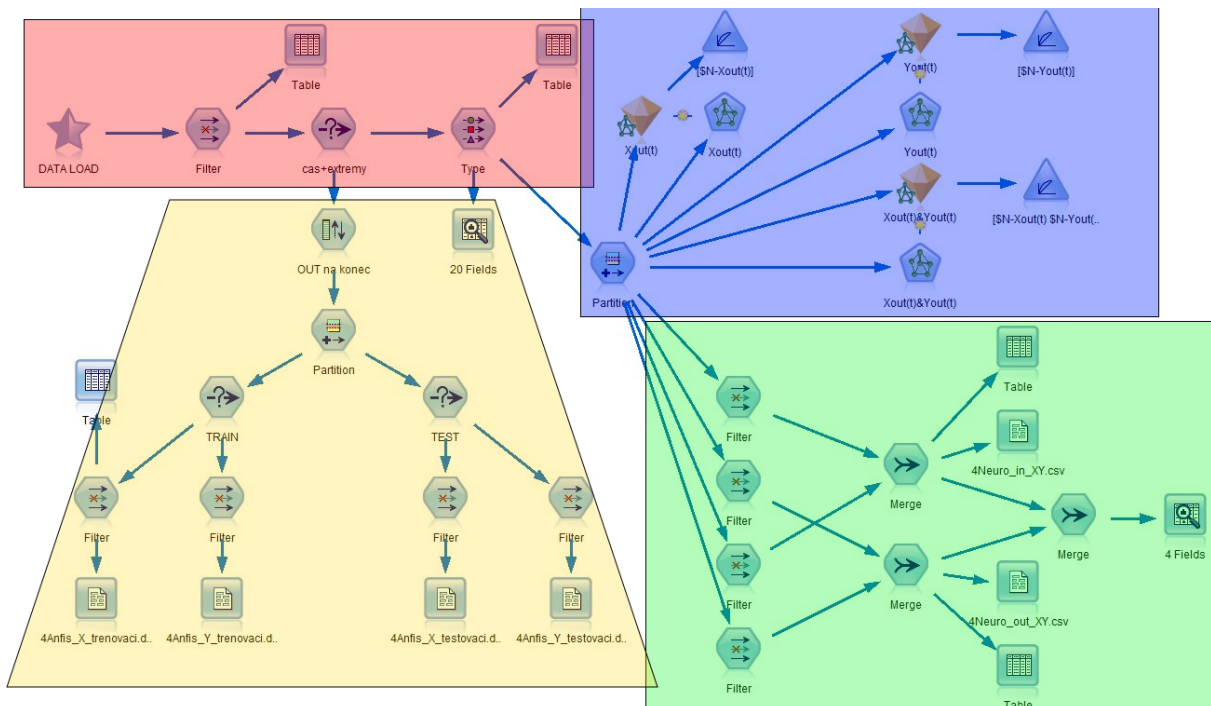
Výstupem tohoto procesu je tedy 20 souborů s daty o logovaných regulačních pochodech ve formátu CSV.

## 2.2 Kompletace souboru dat pro učení CI

Jak již bylo v předchozí kapitole naznačeno, po exportu logů z regulačních pochodů se dále data nezpracovávala v prostředí MATLAB. Zvolil se nástroj IBM SPSS Modeler. Tento software se používá primárně pro data mining, nicméně pro svou velmi dobrou schopnost manipulace s daty se použil pro kompletaci datových souborů, použitých pro učení systémů CI. Následně se také tento nástroj použil pro ověření vhodnosti dat pro učení CI. V prostředí IBM SPSS Modeler se pro analýzy a manipulace s daty vytváří takzvané „streamy“. Jak název napovídá jde v podstatě o proud dat, který se v jednotlivých uzlech nějakým způsobem zpracovává.

Pro potřeby této práce se zkontruoval stream na obrázku níže (viz Obrázek 19). Schéma streamu bez označení jednotlivých částí a větším rozlišením je zobrazeno v příloze B. V červeně vyznačené části se načtou data do prostředí IBM SPSS Modeler a následně se předzpracují. Přesněji jde o 20 logů, které byly výstupem, popsáním v předchozí kapitole. Zeleně označená část kompletuje data pro potřeby učení neuronové sítě. Žlutě je poté označena část streamu pro kompletaci dat, které využívá pro své učení adaptivní neuro-fuzzy inferenční systém. Tyto dvě kompletace datových souborů jsou rozdílné a probíhají zvlášť. Je tomu tak kvůli tomu, že každý nástroj prostředí MATLAB pro učení různých typů CI má jiné požadavky na strukturu datových souborů, které pro učení využívá.

Modře je na streamu vyznačeno ověřování toho, zda jsou data vhodná pro učení výpočetní inteligence. Tomu se však bude věnovat až kapitola 2.3.

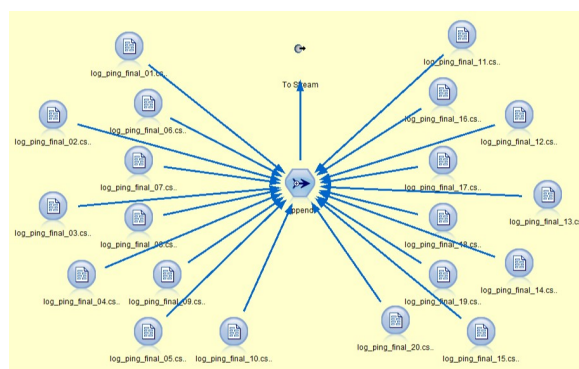


**Obrázek 19:** Schéma v IBM SPSS Modeler pro kompletaci trénovacích dat CI

*Zdroj: vlastní*

### 2.2.1 Načtení a předzpracování dat

Všech 20 logů, tedy datových souborů s průběhem regulačního pochodu, je nutné nejprve načíst do prostředí IBM SPSS Modeler. Toho se docílí pomocí schématu na obrázku níže (viz Obrázek 20). Každý uzel ve schématu představuje načtení jednoho logu regulačního pochodu. Díky transponovaným datovým maticím, které se získaly z prostředí MATLAB, Modeler správně rozpozná atributy a jednotlivé záznamy. Jak je znázorněno ve schématu, všechny jednotlivé logy směřují k uzlu uprostřed. Tento blok se nazývá „append“ a jeho funkcí je sloučit všechny záznamy vstupních souborů do jednoho. Pro zlepšení přehlednosti hlavního streamu je načítání logů shluknuto pod uzel „DATA LOAD“.

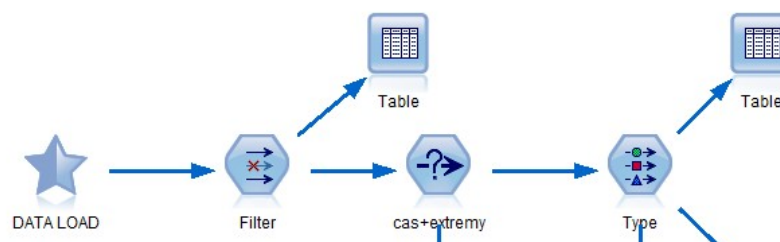


**Obrázek 20:** Načtení jednotlivých logů provedených měření

*Zdroj: vlastní*

Po načtení všech dat probíhá jejich předzpracování znázorněno na schématu níže (viz Obrázek 21). Uzlem „filter“ slouží k pojmenování jednotlivých atributů na názvy veličin, které představují. Následuje uzel „select“, kde se pomocí atributu, který představuje čas, vyfiltrují všechny záznamy, které představují část regulačního pochodu, kdy probíhá kalibrace. Poté se vyfiltrují záznamy, kde mají opožděné veličiny nulovou hodnotu. Vzhledem k množství záznamů není problém s jejich odstraněním. Celkem se tedy odstranily záznamy, které se změřily v časech 0 až 16,5 s. Samotná regulace totiž začínala v čase 16,05 s a všechny opožděné veličiny se staly nenulovými v čase 16,65 s, tedy jednu vzorkovací periodu po čase určeném ke smazání záznamů.

Kromě výše popsaného se v uzlu „select“ odstraňují pro další zpracování dat i záznamy, které mají charakter odlehlých veličin. Jedná se o chybně změřená data, která dosahují hodnot, kterých systém nemohl prakticky dosáhnout. Přesněji dochází k odfiltrování záznamů, kde hodnota akční veličiny či regulační odchylky v obou směrech náklonu plošiny přesahuje interval  $(-1; 1)$ .



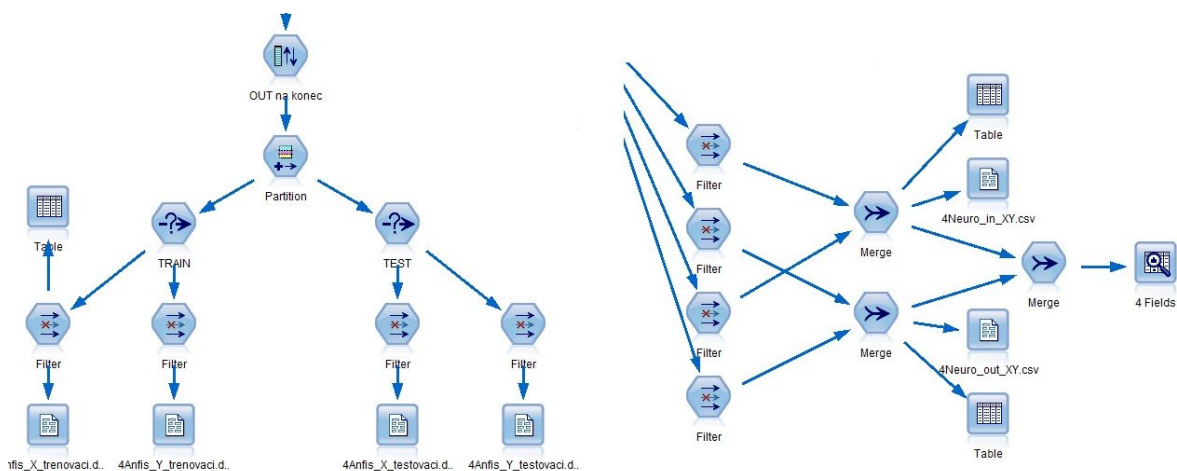
**Obrázek 21:** Načtení a předzpracování dat jednotlivých logů v IBM SPSS Modeler

*Zdroj: vlastní*

Uzlem „type“ se nakonec upřesní typ veličin a jejich role ve výchozím nastavení případných uzlů pro analýzy. Pro časový atribut se tedy nastavilo, aby se do analýz nepřidával, hodnoty akční veličiny v čase  $k$  se nastavily pro osu  $x$  a  $y$  jako výstupy a zbytek jsou vstupy. Všem atributům se nastavil spojitý typ dat.

### 2.2.2 Vytvoření samotných souborů

Jak je zmíněno v úvodní kapitole o kompletaci dat, rozdílné nástroje pro učení CI v MATLABu vyžadují různou strukturu datového souboru pro učení. Práce se v pozdějších kapitolách zabývá učením systému ANFIS a neuronové sítě. Dokončení kompletace dat pro učení CI se tedy provede dvojím způsobem (viz Obrázek 22).



**Obrázek 22:** Vytvoření souborů dat pro učení CI v prostředí IBM SPSS Modeler

*Zdroj: vlastní*

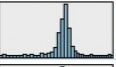
V levé části výše uvedeného schématu (viz Obrázek 22) se vytváří data pro učení adaptivního neuro-fuzzy inferenčního systému. MATLAB požaduje, aby data pro trénování a testování byla v jiných souborech. Protože nástroj v MATLABu dokáže navrhnou ANFIS pouze jako MISO systém, je nutné navrhnout regulátor pro řízení každé osy zvlášť. Z předchozích dvou vět je tedy jasné, že Modeler musí pro učení ANFISu vyexportovat 4 datové soubory. Podmínkou také je, aby byl v souboru atribut odhadovaného výstupu na poslední pozici napravo. Toho se dosáhne prvním uzlem ve schématu výše. Poté se data rozdělí do trénovací a testovací množiny a uzlem „select“ jsou rozděleny do příslušných souborů. Ještě před exportem se uzlem „filter“ označí atributy, které se do exportovaných souborů zapíší.

V pravé části výše uvedeného schématu (viz Obrázek 22) se zpracovávají data pro učení neuronové sítě pro regulaci modelu CE151. Nástroj v MATLABu pro návrh neuronové sítě není omezen pouze jedním výstupem, tudíž stačí navrhnout pouze jednu NS typu MIMO se dvěma výstupy k řízení akční veličiny regulačního obvodu. K návrhu NS je zapotřebí z prostředí Modeler vyexportovat dva soubory. Jeden soubor představuje odhadované akční veličiny, tedy predikanty, a ve druhém budou data, podle kterých se bude odhadovat, prediktory. Před exportem data ve streamu projdou pouze filtrem, který určí atributy pro zápis do souborů a poté se uzlem „merge“ spojí atributy, které se vztahují k řízení náklonu plošiny v ose  $x$  a ose  $y$ .

### 2.2.3 Charakteristiky zpracovávaných dat

Prostředí IBM SPSS Modeler se využilo i pro charakterizování zpracovávaných dat. Jak bylo již naznačeno v závěru kapitoly 2.1.2, celkem se zpracovávalo 20 atributů. Na obrázku níže (viz Obrázek 23) jsou však uvedeny charakteristiky pouze pro 4 atributy. Je tomu tak kvůli

tomu, že ostatní atributy jsou s popsány totožné, pouze zpožděné o různá  $T_s$ . Zobrazené charakteristiky se vztahují k již předzpracovaným datům. Vidíme tedy mimo jiné hodnoty minima, maxima, směrodatné odchylky, počet záznamů a histogram, díky kterému lze tvrdit, že data mají přibližně normální rozdělení.

Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
Xin(t)		Continuous	-0.967	0.958	-0.019	0.266	0.287	--	3786
Yin(t)		Continuous	-0.958	0.958	-0.013	0.269	0.091	--	3786
Xout(t)		Continuous	-0.990	0.960	-0.062	0.236	0.573	--	3786
Yout(t)		Continuous	-0.961	0.996	-0.047	0.235	0.347	--	3786

**Obrázek 23:** Charakteristiky datového souboru pro učení CI

*Zdroj: vlastní*

V celém streamu se pracovalo s atributy:  $X_{in}(t)$ ,  $Y_{in}(t)$ ,  $X_{in}(t-1)$ ,  $Y_{in}(t-1)$ ,  $X_{in}(t-2)$ ,  $Y_{in}(t-2)$ ,  $X_{in}(t-3)$ ,  $Y_{in}(t-3)$ ,  $X_{in}(t-4)$ ,  $Y_{in}(t-4)$ ,  $X_{out}(t)$ ,  $Y_{out}(t)$ ,  $X_{out}(t-1)$ ,  $Y_{out}(t-1)$ ,  $X_{out}(t-2)$ ,  $Y_{out}(t-2)$ ,  $X_{out}(t-3)$ ,  $Y_{out}(t-3)$ ,  $X_{out}(t-4)$ ,  $Y_{out}(t-4)$ . První písmeno udává, jaké osy náklonu plošiny se veličina týká. „In“ či „out“ poté popisuje, zda se jedná o vstup nebo výstup regulátoru. V závorce je poté uvedeno, o kolik  $T_s$  jsou data zpožděna. Pro lepší přehlednost jsou atributy detailně popsány v tabulce níže (viz Tabulka 1).

**Tabulka 1:** Doplnující informace o zpracovávaných datech

Název atributu	Popis
$X_{in}(t)$	První vstup do regulátoru, tedy naměřená regulační odchylka mezi cílem regulace a polohou kuličky na plošině v ose $x$ .
$Y_{in}(t)$	Druhý vstup do regulátoru, tedy naměřená regulační odchylka mezi cílem regulace a polohou kuličky na plošině v ose $y$ .
$X_{out}(t)$	První výstup z regulátoru, tedy akční veličina, která řídí náklon plošiny v ose $x$ .
$Y_{out}(t)$	Druhý výstup z regulátoru, tedy akční veličina, která řídí náklon plošiny v ose $y$ .

*Zdroj: vlastní*

## 2.3 Ověření kvality trénovacích dat

Protože celý proces návrhu regulátorů na bázi CI je časově poměrně náročný, zkonstruoval se v rámci této práce systém pro ověření toho, zda je vhodné naměřená data pro toto učení výpočetní inteligence použít. Přesněji se jedná o testovací neuronovou síť v prostředí IBM SPSS Modeler. Tuto NS nelze použít přímo pro řízení regulované soustavy, nicméně lze ji využít pro analýzu trénovacích dat k posouzení jejich kvality. Poskytne tedy informaci o tom,

s jakou přesností se na základě vstupních dat dokáží predikovat výstupy. Pokud je přesnost malá, nemá smysl taková data dále používat pro návrh regulátorů na bázi CI.

Ověřování dat se provedlo pomocí modré části schématu v úvodu kapitoly 2.2 (viz Obrázek 19). Pro ověření dat k učení systému ANFIS se vytvořily dvě neuronové sítě pro řízení náklonu plošiny do každé osy. Pro učení NS, které bude regulovat model CE151 se vytvořila pouze jedna testovací NS se dvěma výstupy. Na začátku ověřování se jako vstupy zvolily všechny možné, poté se NS postupně zjednodušovala a počet vstupů se redukoval. Proces tohoto ověřování je detailněji zachycen v tabulce níže (viz Tabulka 2). Sloupec „Vstupy“ označuje počet prediktorů. První číslo se vztahuje k počtu opožděných regulačních odchylek, druhé potom k počtu opožděných akčních veličin. Číslo 3 tedy například znamená, že se jako prediktory použily data z časů  $k$ ,  $k-1$ ,  $k-2$  a  $k-3$ . Druhé číslo začíná až od jedné, protože akční veličina v čase  $k$  je predikantem.

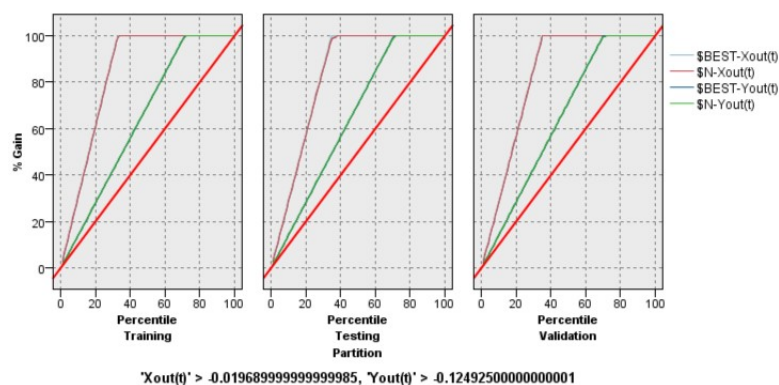
**Tabulka 2:** Přesnost predikce ověřovací NS v závislosti na různých vstupech

Vstupy	Přesnost [%]	Vstupy	Přesnost [%]	Vstupy	Přesnost [%]
0 -	54,5	1 1	95,0	2 2	99,7
0 1	83,6	1 2	96,0	3 2	99,7
0 2	86,4	1 3	96,2	2 3	99,8
0 3	88,0	1 4	96,5	4 1	99,8
0 4	88,1	3 3	99,6	4 2	99,8
1 -	88,9	4 4	99,6	2 1	99,8
3 -	89,2	2 4	99,6	3 4	99,9
2 -	89,4	3 1	99,6		
4 -	89,8	4 3	99,6		

*Zdroj: vlastní*

Z této tabulky tedy vyplývá, že největší přesnost má ověřovací neuronová síť v případě, že se jako vstupy použijí veličiny  $e(k)$  opožděné o  $0T_s$ ,  $1T_s$ ,  $2T_s$  a  $3T_s$  a  $u(k)$  opožděné o  $1T_s$ ,  $2T_s$ ,  $3T_s$  a  $4T_s$ . Protože je ale vhodné vzhledem k nárokům na výkon počítače použít pro regulaci co nejjednodušší NS, navrhne se CI regulátor se vstupy  $e(k)$ ,  $e(k-1)$ ,  $e(k-2)$  a  $u(k-1)$  a to v obou osách řízení plošiny. Takový regulátor má přesnost horší pouze o 0,1 % od nejlepšího možného, což je zanedbatelné. Navíc se místo šestnácti vstupů navrhne regulátor pouze s osmi, čímž výrazně klesnou nároky na výpočetní výkon. Takto navržený regulátor bude také pracovat v souladu s teorií z kapitoly 1.3.

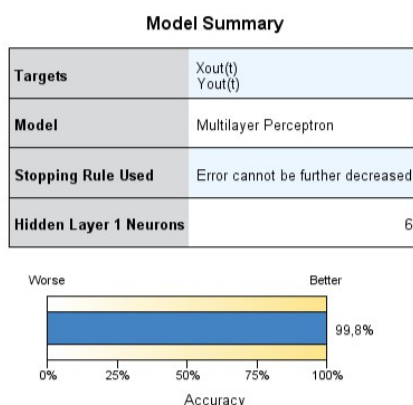
Pro ověření toho, zda nedošlo k přeučení ověřovací neuronové sítě o osmi vstupech se využilo evaluačního grafu (viz Obrázek 24). Protože křivky samotné evaluace splývají s nejlepšími možnými v každé množině dat, můžeme tvrdit, že k přeučení nedošlo a predikce má opravdu vysokou přesnost.



**Obrázek 24:** Evaluační graf ověřování dat pro učení CI pomocí neuronové sítě

*Zdroj: vlastní*

Jak již bylo naznačeno výše, ANFIS regulátory se navrhnu dva, jeden pro řízení každé osy náklonu plošiny. Z analýzy navržené ověřovací NS vyplývá, že počet neuronů ve skryté vrstvě pro ANFIS regulátor bude roven dvěma. Stejně tak z analýzy výsledků pro regulátor na bázi NS vyplývá, že počet neuronů ve skryté vrstvě je roven šesti (viz Obrázek 25). Protože ale na ANFIS stačí dva, zkusí se použít na NS regulátor v rámci zmenšování nároků na výkon 4. Pokud bude rozdíl v přesnosti minimální, použije se nižší počet neuronů.

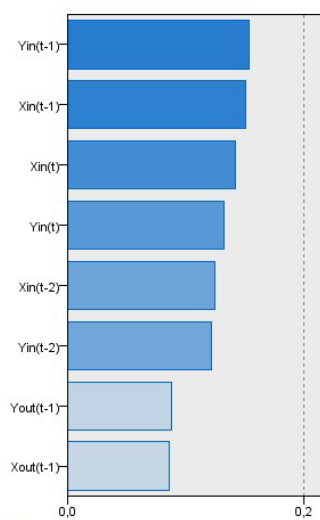


**Obrázek 25:** Nejlepší ověřovací model neuronové sítě pro dva výstupy

*Zdroj: vlastní*

Pro lepší představu toho, jaká veličina má na predikci největší podíl se v prostředí IBM SPSS Modeler sestrojil graf uveden níže (viz Obrázek 26). Je zřejmé, že pro obě osy řízení platí velmi podobné zákonitosti. Pořadí vlivu veličin na predikci od největšího je u nich společné. Největší podíl na řízení má regulační odchylka, opožděná o  $1T_s$ , dále následuje neopozděná regulační odchylka v čase  $k$ . Těsně za touto veličinou se pohybuje  $e(k-2)$  a nakonec, s poměrně větším rozdílem v míře ovlivňování predikce, se vyskytuje opožděná akční veličina o  $1T_s$ . Podle tohoto grafu by se teoreticky daly odhadnout stavitelné parametry  $q_0$ ,  $q_1$  a  $q_2$  v rovnici (5) kapitoly 1.3, která popisuje řízení soustavy diskretním PID regulátorem.





**Obrázek 26:** Míra významu prediktorů ověřovací neuronové sítě

*Zdroj: vlastní*

### 3 NEURONOVÁ SÍŤ JAKO ŘÍDÍCÍ SYSTÉM

Tato část práce se zaměří na návrh regulátoru, který pracuje na bázi CI, přesněji systému neuronové sítě. Vychází se zde z obou předchozích kapitol. V úvodu je teoreticky popsáno vymezení pojmů a fungování neuronových sítí. Poté se práce zaměří na návrh regulátoru v praxi pro model CE151, kde plynule naváže na dřívější poznatky a výstupy z kapitoly 2. Nakonec dojde k jeho implementaci a zhodnocení toho, jak věrně dokáže napodobit původní regulační pochod.

#### 3.1 Teoretický úvod do neuronových sítí

Ještě před popisem samotných neuronových sítí je vhodné stručně definovat pojem soft computing, pod který samotná problematika neuronových sítí spadá. Tento pojem také úzce souvisí s dalšími kapitolami práce, které pojednávají o fuzzy logice. Soft computing je inovativní přístup ke konstruování systémů výpočetní inteligence. Nyní je již obecně přijato, že komplexní problémy reálného světa vyžadují inteligentní systémy, které kombinují znalosti, techniky a metodologie z různých vědních oblastí. Tyto inteligentní systémy by se měly být schopny přizpůsobovat a učit se ve stále se měnícím prostředí a také by měly být schopné objasnit, jak se rozhodují při řešení problémů. [5]

Dle [5] se soft computing vyznačuje lidskou úrovní expertízy, výpočetní modely jsou inspirovány biologií, objevují se tu nové optimalizační techniky, numerické výpočty oproti symbolickým, nové domény aplikovatelnosti, proces učení probíhá bez modelů, větší výpočetní náročností, tolerancí k neurčitosti a chybám, aplikovatelností v reálném světě a charakteristikami řízenými cílem. V tabulce níže (viz Tabulka 3) jsou uvedeny složky soft computingu včetně porovnání jejich hlavních výhod.

**Tabulka 3:** Složky soft computingu

Metodologie	Hlavní výhody
Neuronové sítě	Učení a adaptace
Teorie fuzzy množin	Reprezentace vědomostí pomocí fuzzy IF-THEN pravidel
Genetické algoritmy a simulované žíhání	Systematicky náhodné vyhledávání

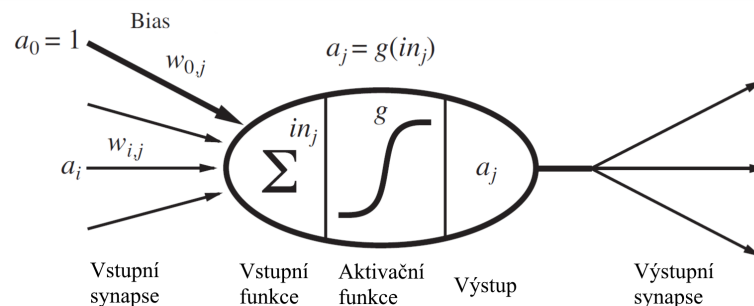
*Zdroj: upraveno dle [5]*

Neuronová síť je masivně paralelní procesor, složený z jednoduchých výpočetních jednotek, které mají přirozený sklon k ukládání zkušenostních znalostí a jejich poskytování k dalšímu použití. Umělá neuronová síť připomíná lidský mozek ve dvou aspektech:

- Síla spojení mezi neurony, také známá jako váha synapse, je využívána pro ukládání získané znalosti.
- NS získává znalost ze svého prostředí pomocí procesu učení. [2]

Proces učení probíhá většinou metodou zpětného šíření chyby (backpropagation), kde dochází k postupné úpravě vah synapsí mezi neurony. Pro návrh NS pro model CE151 se využije tzv. učení s učitelem (supervised). Jde o způsob učení, kdy NS dostane výstupní data k příslušné kombinaci dat vstupních. Po procesu učení je nutné zkontrolovat, zda se NS nepřeučila. Taková síť by se naučila „z paměti“ všechny kombinace vstupů a výstupů datového souboru, na základě kterého se učila a nebyla by schopna generalizace. Pro ověření přeučení se data rozdělují většinou do dvou množin, kdy na první probíhá učení a na druhé ověření přesnosti NS. Uvádí se, že NS by se měla učit na 2/3 všech dostupných dat pro učení. [10]

Na obrázku níže (viz Obrázek 27) je znázorněna již zmiňovaná výpočetní jednotka neuronové sítě, tedy neuron. Každý takovýto neuron se aktivuje, pokud je překročena určitá prahová hodnota, která se počítá ze vstupních synapsí. Přesněji z vah vstupních synapsí, ve kterých je uložena určitá znalost NS a výstupů neuronů, které daný neuron předchází. Na neuron také působí tzv. *bias*. Na obrázku je zachycen pomocí parametru  $a_0$ , což je konstanta a váhou synapse  $w_{0,j}$ , z které vyplývá, že bias není propojen z předešlými neurony. Pomocí biasu se nastavují parametry aktivační funkce. [12]



**Obrázek 27:** Jednoduchý model neuronu

*Zdroj: upraveno podle [12]*

Výstup aktivního neuronu se definuje rovnicí (8) níže:

$$a_j = g(in_j) = g\left(\sum_{i=0}^n w_{ij} a_i\right), \quad (8)$$

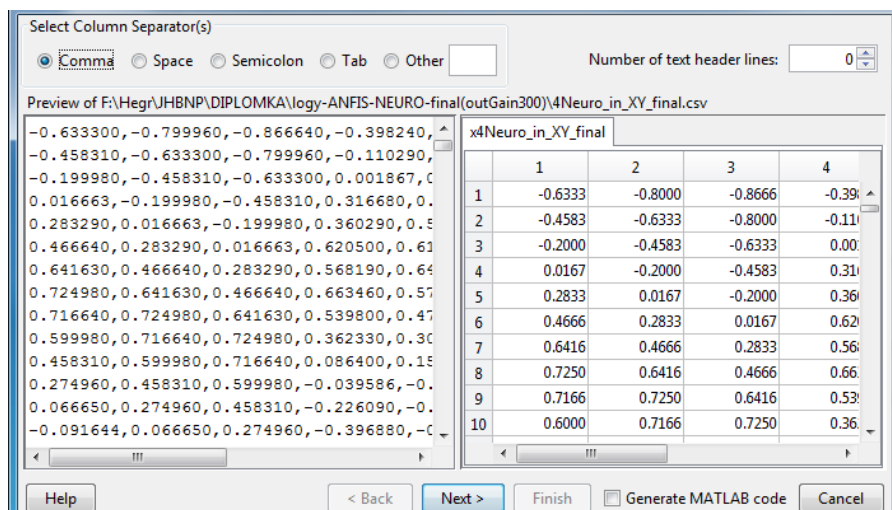
kde:  $a_j$  je výstupní hodnota daného neuronu,  $g$  je aktivační funkcí neuronu,  $w_{ij}$  je váha synapse mezi  $i$ -tým a  $j$ -tým neuronem,  $a_i$  je výstupní hodnota předešlého,  $i$ -tého neuronu. [12]

## 3.2 Návrh neuronové sítě pro regulaci modelu CE151

Jak už vyplývá z předchozí kapitoly, neuronové sítě se dá využít jako regulátoru pro systém operující v reálném čase. Dalo by pomoci ní regulovat i spojitě, nicméně pro potřebu modelu CE151 postačí regulace diskretní, a to s původní vzorkovací periodou  $T_S=0,15$  s. V této části se práce zaměří na návrh NS pomocí dříve získaného souboru dat, kterým se bude NS učit. Výstupem procesu učení bude pouze neuronová síť, ne samotný regulátor. Tato NS se bude muset dále implementovat jako část regulátoru, čímž se bude zabývat kapitola 3.4.

Pro tvorbu neuronových sítí prostředí MATLAB nabízí poměrně propracovaný nástroj s názvem Neural Network Fitting App. Tento nástroj se spustí pomocí příkazu „nnstart“ v hlavním pracovním okně MATLABu. Otevře se nové okno s nabídkou pro výběr typu navrhované NS, kde se zvolí „Fitting App“. Pro účely této práce se využije tato volba, nicméně nástroj dále nabízí prostředí pro návrh NS pro rozeznávání vzorů, shlukování, nebo práci s časovými řadami.

Prvním krokem je do pracovní paměti nástroje nainportovat data, pomocí kterých se bude NS učit. Jak již bylo předznamenáno v kapitole 2.2.2, nainportovat se musí přesněji 2 datové soubory. První soubor obsahuje prediktory, druhý predikanty. Jednotlivé atributy jsou odděleny čárkou a pro oddělení desetinného čísla se používá tečka. Proces importu usnadňuje jednoduchý průvodce, který vizualizuje všechna nastavení v reálném čase (viz Obrázek 28).

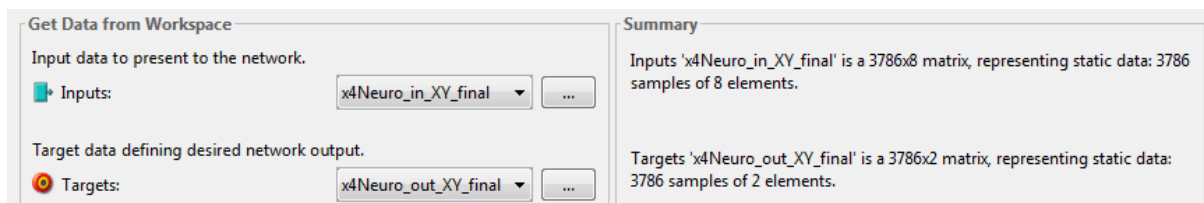


Obrázek 28: Import souboru dat pro učení neuronové sítě

Zdroj: vlastní

Po úspěšném importu dat se v průvodci návrhem NS z pracovní paměti označí prediktory jako vstupy (Inputs) a predikanty jako odhadované výstupy (Targets). V pravé části okna (viz Obrázek 29) se poté zobrazí, kolik atributů (elements) je označeno jako vstup a kolik jako výstup včetně počtu jejich záznamů (samples). V případě dat pro učení NS pro model CE151







se využívá 8 atributů jako vstup (prediktor) a 2 atributy jako výstup (predikant). Od toho se budou odvíjet i počty neuronů na vstupu a výstupu NS. V rámci těchto atributů je pro učení NS k dispozici 3786 záznamů, které představují různé kombinace hodnot vstupních a výstupních atributů.



**Obrázek 29:** Vlastnosti použitých souborů dat pro učení neuronové sítě

*Zdroj: vlastní*

V dalším kroku se zvolí počet neuronů ve skryté vrstvě neuronové sítě. V kapitole 2.3 se navrhla možnost zvolit počty 4 a 6. Realizoval se tedy návrh pro obě varianty a závěrem bylo, že 4 neurony ve skryté vrstvě NS jsou více než dostačující. Nejen, že klesly nároky na výkon počítače simulujícího regulátor, ale NS se čtyřmi neurony ve skryté vrstvě pracuje s vyšší přesností. Tato přesnost je spolu s dalšími charakteristikami výsledků učení uvedena na obrázku níže (viz Obrázek 30).

Results			
	 Samples	 MSE	 R
 Training:	2650	5.51076e-10	9.99999e-1
 Validation:	757	5.94775e-10	9.99999e-1
 Testing:	379	5.03818e-10	9.99999e-1

**Obrázek 30:** Výsledky učení neuronové sítě

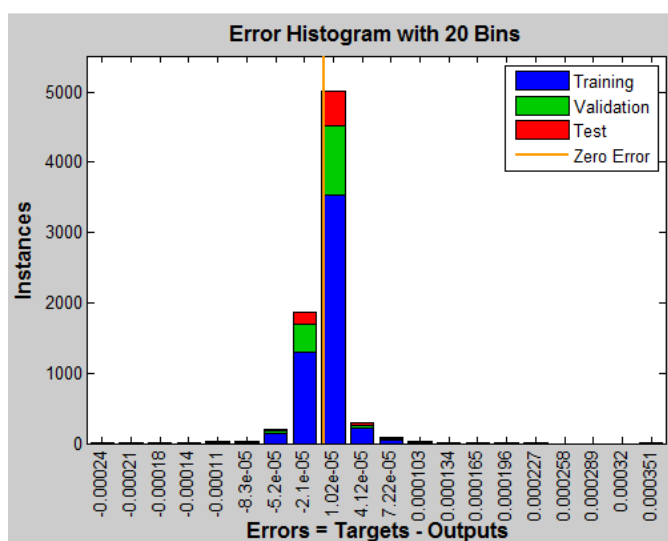
*Zdroj: vlastní*

Výše uvedené charakteristiky výsledků učení NS udávají v každém řádku výsledky pro každou množinu dat. V kapitole 3.2 bylo uvedeno, že se pro učení má využít 2/3 dostupných dat. Tyto 2/3 dat představuje trénovací množina (training), u které se však v tomto případě zvolil poměr 7/10. Zbylé 3/10 dat (množiny validation a testing) se používají na ověření toho, zda není NS přeučená a je schopná generalizace výpočetního problému. V prvním sloupci je zobrazen počet záznamů dané množiny, druhý sloupec udává hodnotu střední kvadratické chyby a poslední sloupec koeficient korelace prediktorů a predikantů.

Z těchto výsledků vyplývá, že prostředí MATLAB bylo schopné navrhnout NS, která pracuje s mnohem vyšší přesností než ověřovací NS z prostředí IBM SPSS Modeler. Střední kvadratická chyba se pohybuje kolem hodnoty  $5,5 \times 10^{-10}$ , což v přepočtu znamená, že chyba navržené NS je 0,000000055 %. Korelace mezi vstupy a výstupy je také velmi dobrá.

Maximální hodnota  $R$  může být 1, tato NS má  $R$  rovno 0,9999. Tento koeficient korelace můžeme umocnit a získáme koeficient determinace. Díky němu můžeme tvrdit, že neuronová síť je schopna vysvětlit téměř 100 % hodnot predikovaných veličin.

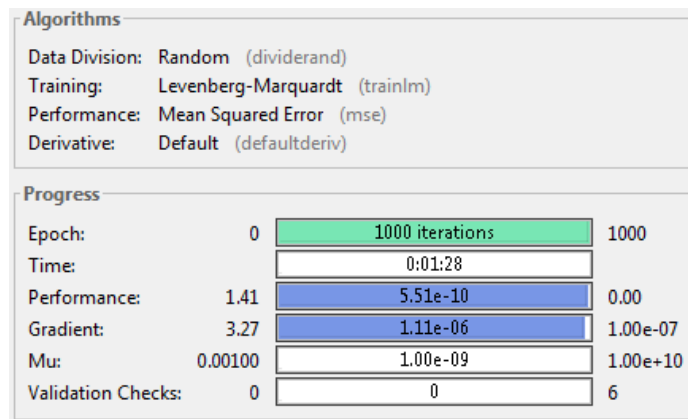
Pro detailní pohled na chyby NS prostředí MATLAB poskytlo jejich histogram (viz Obrázek 31). Z tohoto grafu jasně vyplývá, že většina chyb se v průběhu učení NS pohybovala kolem nuly, přesněji v intervalu  $(-2,1 \times 10^{-5}; 1,02 \times 10^{-5})$ . Můžeme také říci, že chyby mají velmi podobné rozdělení pravděpodobnosti v rámci všech množin. Díky tomu lze jasně konstatovat, že NS není přeučená a bude schopna správně reagovat i na hodnoty vstupů, s kterými se nesešla během procesu jejího učení.



**Obrázek 31:** Histogram chyb po učení neuronové sítě

*Zdroj: vlastní*

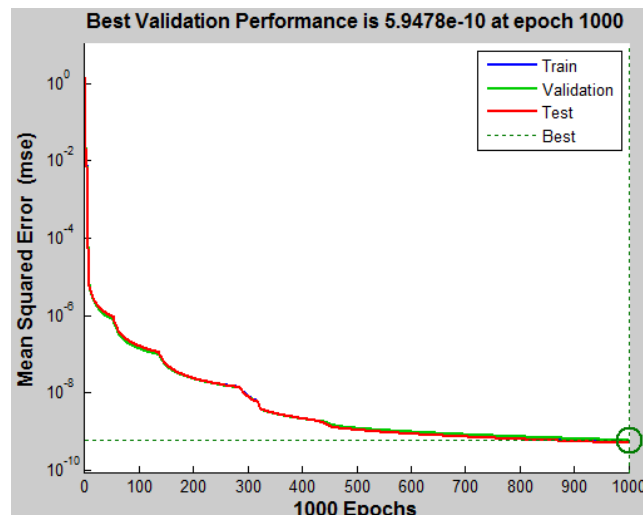
Níže uvedený obrázek zachycuje další detaily průběhu učení NS. Pro rozdělení dat do trénovací, testovací a validační množiny se využilo funkce „dividerand“, které rozdělí data náhodně. Pro trénování, přesněji optimalizaci synaptických vah se využilo Levenberg-Marquardtova algoritmu. Pro ohodnocení kvality navrhované NS se použila střední kvadratická chyba mezi výstupy NS a výstupy (predikanty) z trénovacího souboru dat, kterou se systém snažil minimalizovat. Dále dostáváme informaci o tom, že se NS optimalizovala v tisíci iteracích a čas jejího naučení byl téměř 1,5 minuty.



**Obrázek 32:** Detaily průběhu učení neuronové sítě

*Zdroj: vlastní*

Průběh učení neuronové sítě také můžeme sledovat v závislosti střední kvadratické chyby na dané iteraci optimalizačního cyklu (viz Obrázek 33). Jak je z níže uvedeného grafu patrné, v průběhu několika prvních desítek iterací se docílilo značné minimalizace střední kvadratické chyby. Učení by pravděpodobně mohlo skončit v iteraci 500, nicméně algoritmus optimalizace je navržen tak, aby minimalizoval chybu i jen o nepatrné hodnoty. Dá se předpokládat, že proces učení by pro nejlepší možný výsledek přesáhl 1000 iterací, nicméně toto se v průvodci návrhem NS v MATLABu nastavit nemohlo a je to i poměrně zbytečné vzhledem k chybě, která je v řádech  $10^{-10}$ .

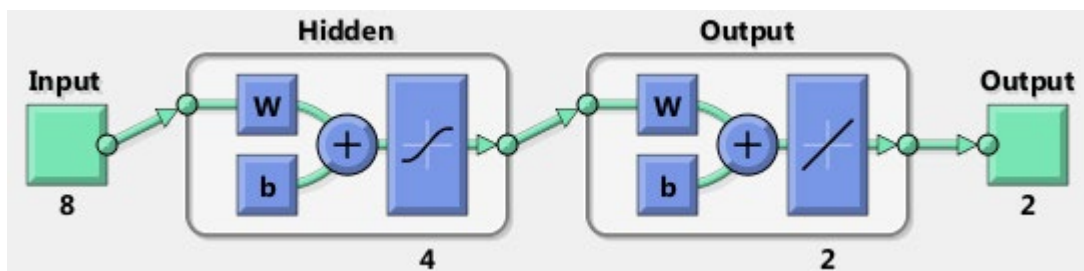


**Obrázek 33:** Průběh učení neuronové sítě v grafu

*Zdroj: vlastní*

Struktura výsledné neuronové sítě je schematicky naznačena na obrázku níže (viz Obrázek 34). Vidíme, že NS má 8 vstupů a 2 výstupy. Počet neuronů v tzv. skryté vrstvě je nakonec 4 a výstupní vrstva NS obsahuje neurony 2. Pro skrytou vrstvu se v neuronech použila sigmoidální aktivační funkce, ve výstupní se použila funkce lineární. Písmena  $w$  označují synaptické váhy mezi neurony,  $b$  označuje bias vrstvy neuronů. Jak již bylo zmíněno výše,

v průběhu návrhu NS se určí pouze počet neuronů ve skryté vrstvě. Počet vstupů a výstupů záleží na struktuře trénovacích dat. Vidíme také, že se jedná o síť dopřednou (feedforward), protože schéma neobsahuje žádné zpětné vazby a cykly. [10][11]



**Obrázek 34:** Schéma struktury neuronové sítě pro řízení modelu

*Zdroj: vlastní*

V závěrečné fázi dává průvodce návrhem NS na výběr, zda chceme výstup vyexportovat do různých formátů nebo pokračovat optimalizací právě navržené NS. Protože střední kvadratická chyba je v našem případě opravdu malá, je vhodné síť vyexportovat. Celá simulace regulátoru probíhá v prostředí Simulink, proto se zvolí i tento typ exportovaného souboru, s kterým se bude pracovat dále.

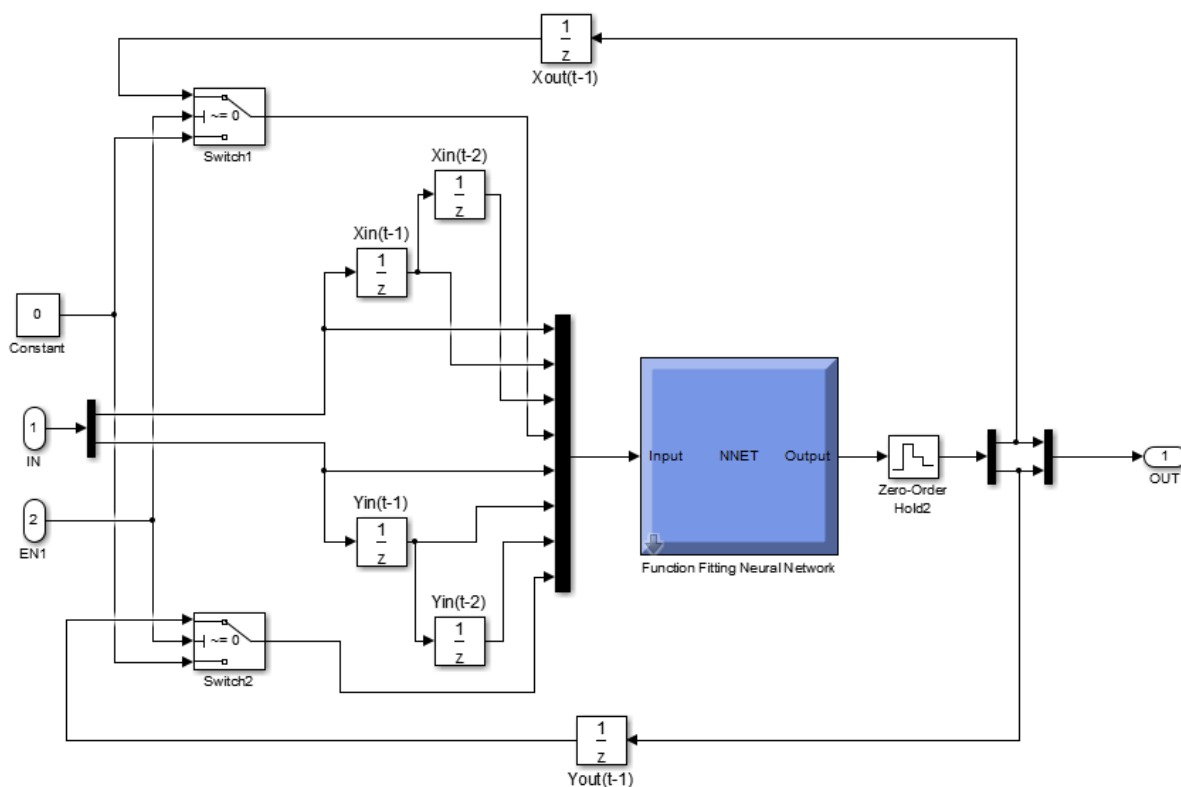
### 3.3 Implementace neuro-regulátoru a jeho zhodnocení

Jak již bylo poznamenáno v předchozí kapitole, pouhý vyexportovaný soubor z návrhu NS pro regulaci modelu CE151 nestačí. V této fázi je nutné vrátit se do původního schématu regulačního systému a připravit vhodné prostředí pro implementaci nové NS. Pro větší přehlednost se toto prostředí přesunulo dovnitř jednoduchého bloku v obecném schématu (viz Obrázek 45 – oranžově označená část) a představuje vše kromě samotného bloku představujícího neuronovou síť. Detailní schéma, které se zkonstruovalo za účelem implementace NS pro regulaci modelu CE151 je znázorněno na obrázku níže (viz Obrázek 35).

Protože je toto schéma umístěno v místě původního PID regulátoru, na vstupu s označením „IN“ přichází do neuro-regulátoru dva signály. Jsou jimi regulační odchylka pro řízení plošiny v ose  $x$  a regulační odchylka pro řízení plošiny v ose  $y$ . Aby se tyto signály rozdělily, je nutné použít blok s označením „demux“. Ve schématu výše je umístěn hned za vstupem „IN“. Takto rozdělený signál můžeme dále zpracovávat. Každou regulační odchylku je dle závěru kapitoly 2.3 nutné rozdělit na 3 části a dvě z nich opozdit o  $T_s$  a  $2T_s$ . Dále je nutné do obvodu zavést zpětnou vazbu z výstupu neuro-regulátoru a opozdit ji o jednu  $T_s$ . Tyto čtyři signály v každé ose řízení náklonu plošiny se konečně mohou přivést spojené na vstup NS. Spojení se provedlo blokem „mux“, který má opačnou funkci jako dříve zmíněný „demux“ a vstupní signály spojí



do jednoho datového proudu. Blokem „OUT“ se signál dostává zpět do obecného schématu, kde pokračuje jako akční veličina, kterou se reguluje soustava.



**Obrázek 35:** Detailní schéma navrženého neuro-regulátoru v prostředí MATLAB-Simulink

*Zdroj: vlastní*

Dále schéma obsahuje několik podpůrných bloků. Jak je již zmíněno v kapitole 1.6.3, spuštění samotné regulace se provádí až po kalibraci soustavy pomocí signálu z bloku simulace, který generuje po 16 s jednotkový skok. Toto opoždění spuštění regulace je realizováno spínači „Switch1“ a „Switch2“. Kvůli redukci neočekávaných chyb se do schématu zavedlo několik „Zero-Order Hold“ bloků, které zajišťují, že signál v průběhu jedné vzorkovací periody zůstane konstantní.

Po návrhu pomocného obvodu a připojení neuro-regulátoru se provedlo 50 měření času stabilizace regulačního obvodu s ping-pongovým míčkem, který se používal jako podklad při návrhu NS. Pro zjištění míry adaptability nového regulátoru se provedlo ještě dalších 10 měření s kuličkou s pogumovaným povrchem a 10 měření s menší, kovovou kuličkou. Výsledky průměrných časů stabilizace jsou uvedeny v tabulce níže (viz Tabulka 4). Tyto časy jsou zmenšeny o 16 s, při kterých probíhala kalibrace soustavy. Vidíme tedy, že neuro-regulátor je schopen napodobit původní způsob regulace, a dokonce má o trochu lepší schopnost adaptability.

**Tabulka 4:** Průměrné časy stabilizace RO pomocí neuro-regulátoru

<b>Typ použité kuličky</b>	<b>Čas stabilizace PID [s]</b>	<b>Čas stabilizace neuro-regulátoru [s]</b>
Ping-pongový míček	22,623	22,577
Kovová s pogumovaným povrchem	17,06	13,915
Kovová	18,06	17,51

*Zdroj: vlastní*

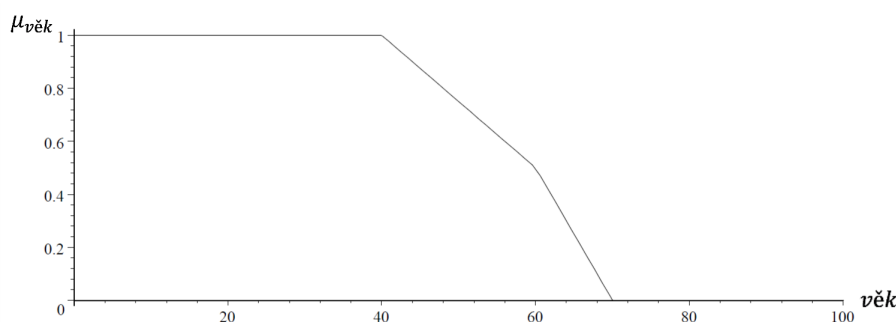
Detailnější analýzou neuro-regulátoru společně s průběhy regulace se bude věnovat kapitola 5, kde se výsledky pro lepší přehlednost rozeberou v porovnání s původním způsobem regulace a dalším regulátorem na bázi CI.

## 4 SPOJENÍ NEURONOVÉ SÍTĚ A FUZZY LOGIKY PRO ŘÍZENÍ SYSTÉMU

Tato část práce je věnována návrhu a implementaci regulátoru pro model CE151 na bázi CI, přesněji systému, který spojuje neuronové sítě a fuzzy logiku v tzv. adaptivní neuro-fuzzy inferenční systém. V úvodu se práce zaměří na teoretický popis fuzzy logiky a poté samotného ANFISu. Protože neuronové sítě jsou rozebrány v kapitole 3, práce se jimi zde zabývat nebude. Po teoretickém úvodu následuje návrh ANFIS regulátoru pro řízení modelu CE151 a jeho implementace. V závěru se vyhodnotí jeho funkčnost a případně i efektivita.

### 4.1 Teoretický úvod do fuzzy množin

Matematické modelování konceptu fuzzy množin představil v roce 1965 L. A. Zadeh. Základem tohoto konceptu je to, že významy přirozené řeči jsou otázkou stupně či míry. Pokud nějakého člověka označíme za mladého, toto tvrzení nemusí být ani pravda ani nepravda. Pokud je věk člověka  $x$ , potom pravda, nebo přesněji „kompatibilita“  $x$  s tímto tvrzením je otázkou stupně nebo míry. Tato míra, přesněji míra příslušnosti může nabývat hodnot v intervalu  $[0; 1]$ . Oproti běžným množinám tedy nějaké tvrzení nenabývá pouze dvou hodnot (pravda a nepravda), ale má určitou míru příslušnosti k pravdě a taktéž k nepravdě. Příklad fuzzy množiny můžeme vidět na obrázku níže (viz Obrázek 36). Samotná křivka se nazývá funkce příslušnosti, osa  $x$  představuje univerzum hodnot, na kterých je fuzzy množina definována a osa  $y$  představuje samotné hodnoty míry příslušnosti. [9]



**Obrázek 36:** Příklad pro funkci příslušnosti fuzzy množiny "mladý člověk"

*Zdroj: upraveno podle [9]*

Nechť  $X = 0$  je klasická množina a  $\mu_A: X \rightarrow \langle 0; 1 \rangle$  nechť je jejím zobrazením. Fuzzy množinou pak budeme nazývat uspořádanou dvojici  $A = (X, \mu_A)$ . Každé  $\mu_A(x) = 0$  můžeme interpretovat tak, že  $x$  do fuzzy množiny  $A$  nepatří. Pokud je hodnota 1, můžeme prohlásit, že  $x$  do  $A$  patří

a pokud  $\mu_A(x) \in (0; 1)$ , potom nelze s jistotou určit, zda  $x$  patří do  $A$ , přičemž velikost  $\mu_A(x)$  je vyjádřením stupně, míry příslušnosti  $x$  k  $A$ . [11]

Pod pojmem fuzzy systém se poté může rozumět takový systém, jehož proměnné (jedna nebo některé z nich) nabývají hodnot (stavů), které nejsou definovány ostrými čísly (reálnými čísly v matematickém smyslu slova), nýbrž jsou definovány slovními hodnotami, tj. jsou definovány fuzzy množinami. Tyto proměnné jsou tedy jazykové proměnné. Každá taková proměnná je definována na odpovídajícím univerzu, které je velmi často intervalem reálných čísel. V tomto speciálním, ale důležitém případě jsou fuzzy množiny vlastně fuzzy čísla. [6]

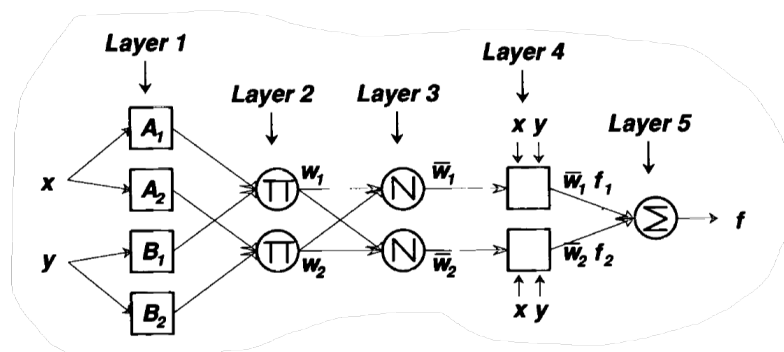
V praxi se teorie fuzzy množin využívá k řízení systémů, které vyžadují ostré (přesné) hodnoty na vstupu a výstupu regulátoru. Takové regulátory musí být doplněny o fuzzifikační a defuzzifikační modul. Regulační odchylka, která přijde na vstup fuzzy-regulátoru je pomocí procesu fuzzifikace převedena na fuzzy množinu. Obdobně defuzzifikace převádí informaci z fuzzy množiny na ostrou hodnotu. Samotný proces regulace poté probíhá pomocí báze pravidel a inferenčního mechanismu, který dle pravidel a vstupních fuzzy množin vyvozuje fuzzy akční veličinu. [6]

## 4.2 Teoretický úvod do problematiky ANFIS

Samotná adaptivní neuronová síť je, jak název napovídá, síťová struktura, která se skládá z určitého počtu uzlů, které jsou propojeny pomocí spojení, ve kterých se signál přenáší pouze jedním směrem. Každý uzel reprezentuje výpočetní jednotku a spojení mezi nimi specifikují jejich kauzální vztah. Všechny nebo část uzlů jsou adaptivní, což znamená, že jejich výstupy závisí na modifikovatelných parametrech, které s těmito uzly souvisí. Pravidlo procesu učení dále udává, jak by tyto parametry měly být měněné, aby se minimalizovala měřená chyba, která je matematickým vyjádřením, které určuje rozdíl mezi skutečným výstupem sítě a požadovaným výstupem. Jinými slovy, adaptivní síť se využívá k identifikaci systému. Touto identifikací se myslí proces určování matematického modelu pro neznámý systém (také cílový systém) pomocí pozorování jeho vstupně-výstupních párů dat. Následujícím snažením je najít vhodnou architekturu sítě a určit parametry, které dokáží nejlépe modelovat neznámý cílový systém, který je popsán právě pouze vstupně-výstupními páry dat. [5]

ANFIS je třída adaptivních sítí, která je funkcionalitou ekvivalentní fuzzy inferenčnímu systému. Celé označení této architektury je fuzzy inferenční systém na bázi adaptivní sítě (adaptive network-based fuzzy inference system) nebo adaptivní neuro fuzzy inferenční systém (adaptive neuro fuzzy inference system), čemuž odpovídá i rozšířené označení ANFIS. [5]

Na obrázku níže můžeme vidět obecnou strukturu systému ANFIS, který je rozdělen do pěti vrstev (Layer 1-5). Tento ukázkový systém má dva vstupy, přičemž každý zpracovávají dvě fuzzy množiny. Pro zjednodušení jsou zde znázorněna pouze 2 pravidla.



**Obrázek 37:** Architektura systému ANFIS pro dvou vstupový FIS typu Sugeno

*Zdroj: [5]*

Ještě před první vrstvou některá literatura uvádí vrstvu, která nijak neupravuje vstupní signál, slouží pouze pro oddělení vstupních veličin. Můžeme ji tedy označit za pomyslnou nultou vrstvu. V první vrstvě se provádí fuzzifikace vstupních ostrých (crisp) hodnot. Každý prvek v této síti je realizovaný neuronem, který představuje jednu fuzzy množinu. Podle počtu neuronů v první vrstvě v rámci jednoho vstupu jsme schopni určit, kolik fuzzy množin tento vstup zpracovává. Druhá a třetí vrstva představuje oblast, kde síť vyhodnocuje fuzzifikované vstupy podle pravidel. Každý neuron čtvrté vrstvy vypočítává výsledné fuzzy hodnoty pravidel, která mají stejný logický závěr (konsekvent). Tyto dílčí výstupy jsou v páté vrstvě spojeny ve výslednou odpověď a defuzzifikovány zpět na ostrou (crisp) hodnotu. [6][11]

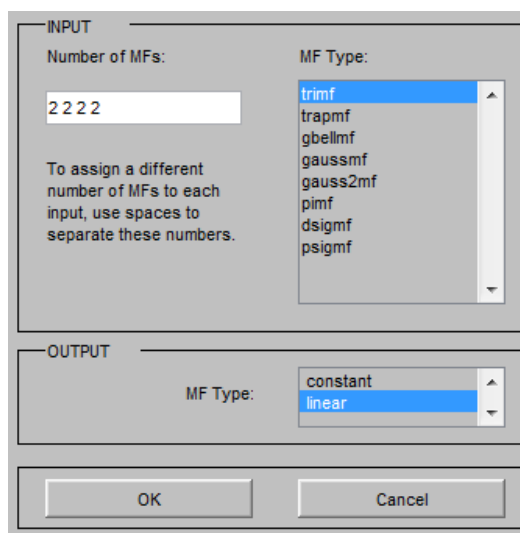
Fuzzy řízení je zdaleka nejúspěšnější aplikace teorie fuzzy množin a fuzzy inferenčních systémů. Díky schopnosti adaptace se ANFIS dá téměř přímo použít do adaptivního řízení a řízení s učení. ANFIS v podstatě může nahradit téměř každou neuronovou síť v řídicím systému a vykonávat stejnou funkci. Nelineárnost a strukturovaná reprezentace znalostí jsou primárními výhodami oproti klasickým lineárním přístupům k adaptivnímu filtrování a adaptivnímu zpracování signálu, jako je identifikace, zpětné modelování, prediktivní kódování, adaptivní vyvažování kanálů, adaptivní filtrace signálu (šum nebo odezva) a tak dále. [5]

### 4.3 Návrh FIS ve spojení s neuronovou sítí pro regulaci modelu CE151

Jak již bylo předznamenáno v dřívějších kapitolách, nástroj MATLABu pro návrh systému ANFIS je omezen pouze na MISO systém. Protože je ale regulátorem nutné řídit dvě akční veličiny, musí se navrhnout systémy dva, které budou pracovat paralelně a nezávisle na sobě.

Následující popis se bude zabývat pouze návrhem jednoho takového systému. Protože zákonitosti ovládání plošiny jsou velmi podobné pro obě osy náklonu, není ani nutné popisovat návrhy obou částí regulátoru. Obě části budou až na zanedbatelné odlišnosti parametrů stejné.

Nástroj v prostředí MATLAB, ve kterém se bude návrh ANFISu provádět se nazývá Neuro-Fuzzy Designer. Spouští se pomocí příkazu „neuroFuzzyDesigner“ nebo kliknutím na příslušnou ikonu v nabídce aplikací MATLABu. Tento nástroj aplikuje techniky fuzzy inference na datové modelování. Tvary funkcí příslušnosti závisí na určitých parametrech, které se oproti manuálnímu návrhu FIS nastavují automaticky. Přesněji se v nástroji Neuro-Fuzzy Designer zvolí tvar funkce příslušnosti, ale její parametry vzhledem k univerzu všech fuzzy množin jedné veličiny se nastaví a optimalizují automaticky. Ukázka nastavování vstupu a výstupu ANFISu pro model CE151 je zobrazena na obrázku níže (viz Obrázek 38).



**Obrázek 38:** Nastavení parametrů pro vygenerovaný FIS

*Zdroj: vlastní*

V rámci tohoto kroku návrhu se vyzkoušely různé počty funkcí příslušností pro jednotlivé veličiny a obdobně také jejich různé tvary (tabulka „MF Type“). Pro výstup se zvolila funkce příslušnosti lineárního typu kvůli tomu, že i při návrhu neuro-regulátoru byly na výstupu neurony s lineární aktivační funkcí. Závěrem této fáze bylo, že každou vstupní i výstupní veličinu charakterizují dvě fuzzy množiny trojúhelníkového tvaru. Toto tvrzení se zakládá na nejmenší průměrné chybě při testování samotného ANFISu a skutečnosti, že je vhodné navrhnout co nejjednodušší systém kvůli menším nárokům na výpočetní výkon. Takto strukturovaný ANFIS měl dokonce o několik řádů menší průměrnou chybu testování než ostatní systémy s vyšším počtem funkcí příslušnosti. O jeho vhodnosti tedy nemá význam pochybovat. Toto nastavení se přesněji vztahuje k fuzzy části systému a pro jeho vytvoření se používá mřížková metoda dělení dat (grid partition). Je možné použít ještě metodu subtraktivního

shlukování (sub. clustering), nicméně k tomu se vztahuje poměrně složité nastavení shlukování. Vzhledem k minimální průměrné chybě testování ANFISu se této metody ani nezkoušelo využít.

Ještě před návrhem struktury ANFIS je vhodné do modelu načíst trénovací a testovací datové soubory. Toho se může docílit buď pomocí externího souboru nebo dat z pracovní paměti MATLABu. Pro větší přehlednost návrhu systému se zvolil externí soubor, jehož popisu se věnuje kapitola 2.2. Pro učení systému ANFIS se využilo pouze trénovací a testovací množiny dat z důvodu bezproblémového učení neuronové sítě pro regulaci modelu CE151, kdy nenastala situace, že by došlo k přeučení. Testovací data tedy postačí k posouzení schopnosti generalizace. Další nastavované parametry pro vytváření systému ANFIS můžeme vidět na obrázku níže (viz Obrázek 39).

Pro optimalizaci se využilo hybridního algoritmu, který je rychlejší a má menší sklony k uvíznutí v lokálním minimu při hledání nejlepšího řešení než metoda zpětného šíření (backpropagation). Tolerance chyby se ponechala z výchozího nastavení na nule. Po nastavení počtu trénovacích epoch, tedy iterací, v kterých dochází k optimalizaci se spustil proces trénování. Protože se po původně nastavených třiceti epochách našla minimální chyba, která se dále nesnižovala, systém se mohl prohlásit za naučený. Jak vidíme na obrázku níže (viz Obrázek 39), chyba se ustálila na hodnotě  $9,7348 \times 10^{-6}$ . Graf představuje chybu systému v jednotlivých epochách optimalizace.

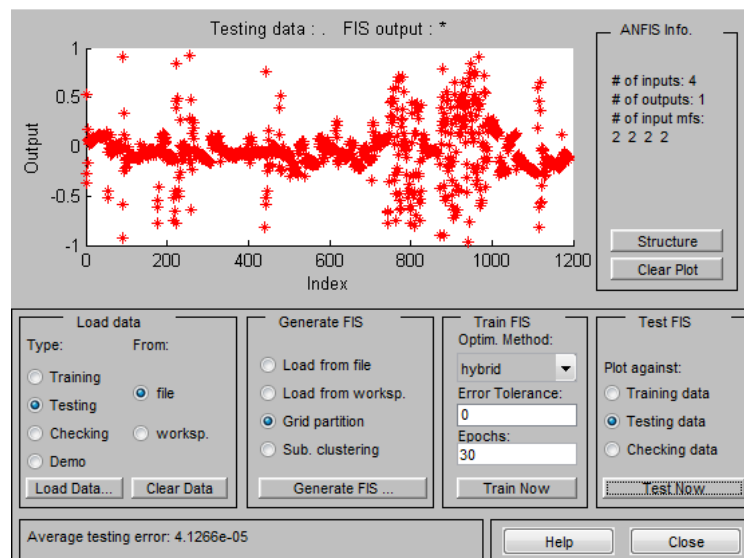


**Obrázek 39:** Nastavení parametrů pro trénování systému ANFIS

*Zdroj: vlastní*

Po naučení systému je potřeba otestovat jeho schopnost generalizace problému podle jeho reakce na hodnoty, s kterými se nesetkal v rámci učení. Tyto odpovědi znázorňuje graf

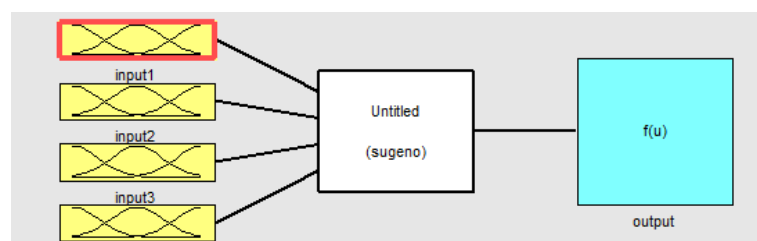
na obrázku níže (viz Obrázek 40). Červené body symbolizují odpověď systému na vstupy testovací množiny dat. Modré body by představovaly vzorový výstup systému z testovacích dat, nicméně vzhledem k vysoké přesnosti systému tyto body překrývají červené. U žádného záznamu nenastala chyba tak velká, aby byla v grafu viditelná. Průměrná odchylka mezi výstupem naučeného ANFISu a vzorovým výstupem v testovacích datech je přibližně  $4,1266 \times 10^{-5}$ , což je hodnota srovnatelná s chybou při trénování. Z toho vyplývá, že k přeučení nedošlo.



**Obrázek 40:** Testování systému ANFIS

*Zdroj: vlastní*

Při zobrazení obecnější struktury ANFISu na obrázku níže (viz Obrázek 41) vidíme, že systém má 4 vstupy, které se zpracovávají v prostředním bloku, kde se nachází inferenční mechanismus a báze pravidel. Výstupem není fuzzy hodnota, jako u fuzzy systémů typu Mamdani, ale funkce vstupních proměnných, což je typické pro fuzzy systémy typu Sugeno. Celkový výstup poté vyjadřuje vážený průměr výstupů (tj. funkcí) pro jednotlivá pravidla. [6]



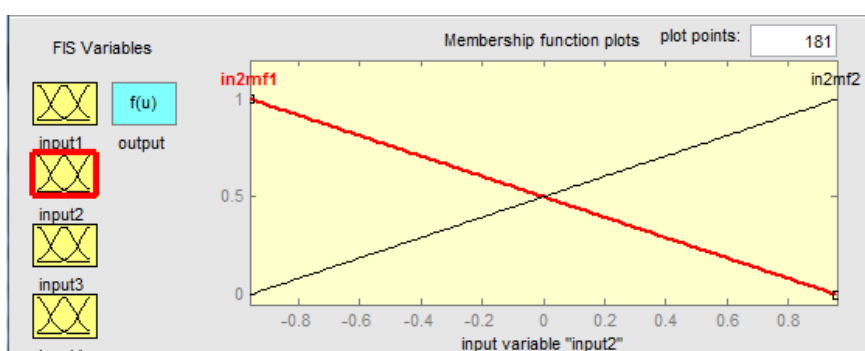
**Obrázek 41:** Struktura vygenerovaného systému ANFIS

*Zdroj: vlastní*

Detail druhého vstupu je vidět na obrázku níže (Obrázek 42). Tento vstup odpovídá regulační odchylce opožděné o  $T_s$ . Je zřejmé, že funkce příslušnosti se protínají přibližně v hodnotě 0. Toto můžeme považovat za ověření správně navržených fuzzy množin. První množina



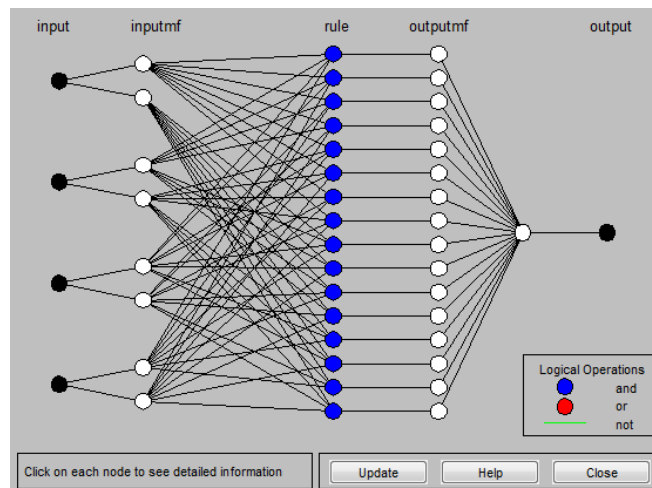
s názvem „in2mf1“ symbolizuje vychýlení kuličky po plošině v jedné ose na jednu stranu. Analogicky fuzzy množina „in2mf2“ představuje vychýlení na druhou stranu. Podle toho, zda se jedná o řízení náklonu plošiny v ose  $x$  nebo  $y$  by tyto množiny bylo možné přejmenovat na „nahore“, „dole“, „vpravo“ a „vlevo“. Pokud by v případě obrázku níže přišla na vstup hodnota např.  $-0,9$ , systém jí přiřadí míru příslušnosti množině „vlevo“ rovnu 1 a příslušnost množině „vpravo“ se bude blížit hodnotě 0. V případě detailní analýzy funkčnosti systému by pak bylo zřejmé, jak a proč dělá systém daná rozhodnutí. Toto je jasná výhoda oproti neuroregulátoru, jelikož tam je báze znalostí zakódovány ve vahách synapsí a nelze tedy jednoznačně určit zákonitosti jeho fungování. Fuzzy množiny na obrázku níže jsou obdobné i pro ostatní vstupy.



**Obrázek 42:** Funkce příslušnosti použité v systému ANFIS

*Zdroj: vlastní*

Neuro-Fuzzy Designer také umožňuje zobrazit detailní strukturu ANFISu v podobě neuronové sítě. Na obrázku níže (viz Obrázek 43) je tato NS zobrazena. Do vrstvy „input“ vstupuje ostrá (crisp) hodnota regulační odchylky. Tato veličina se poté zdvojí a vstupuje do druhé vrstvy neuronů s názvem „inputmf“. Tato vrstva převádí ostré hodnoty vstupů na fuzzy čísla dle příslušných fuzzy množin, probíhá tedy fuzzifikace. Vstupní signál fuzzifikují dva neurony. Každý ohodnocuje míru příslušnosti vstupu k jedné fuzzy množině. Pouhý jeden neuron by takovou operaci nemohl vykonat, jelikož je po fuzzifikaci vyžadována informace o příslušnosti k dvěma fuzzy množinám a jeden neuron má pouze jeden výstup. Další vrstvou je „rule“, tedy báze pravidel. Signál z předchozí vrstvy se rozdělí a vstupuje do jednotlivých pravidel podle toho, zda se v pravidle vyskytuje. Každý neuron pak představuje jedno pravidlo. Celkem se v této vrstvě nachází 16 neuronů, takže můžeme konstatovat, že báze pravidel obsahuje 16 pravidel. Jednotlivé závěry pravidel poté vstupují do vrstvy „outputmf“, kde dochází k jejich ohodnocení výstupní funkcí, spojení ve výslednou odpověď systému a defuzzifikaci. Neuron ve vrstvě „output“ na závěr posílá ostrou hodnotu na výstup celého ANFISu.



**Obrázek 43:** Struktura neuronové sítě v systému ANFIS

*Zdroj: vlastní*

Takto navržený adaptivní neuro-fuzzy inferenční systém je nutné vyexportovat, což nástroj Neuro-Fuzzy Designer umožňuje. Výstupem je pak externí soubor s příponou „fis“, kterého se bude dále využívat v následující kapitole, která se zaměří na implementaci.

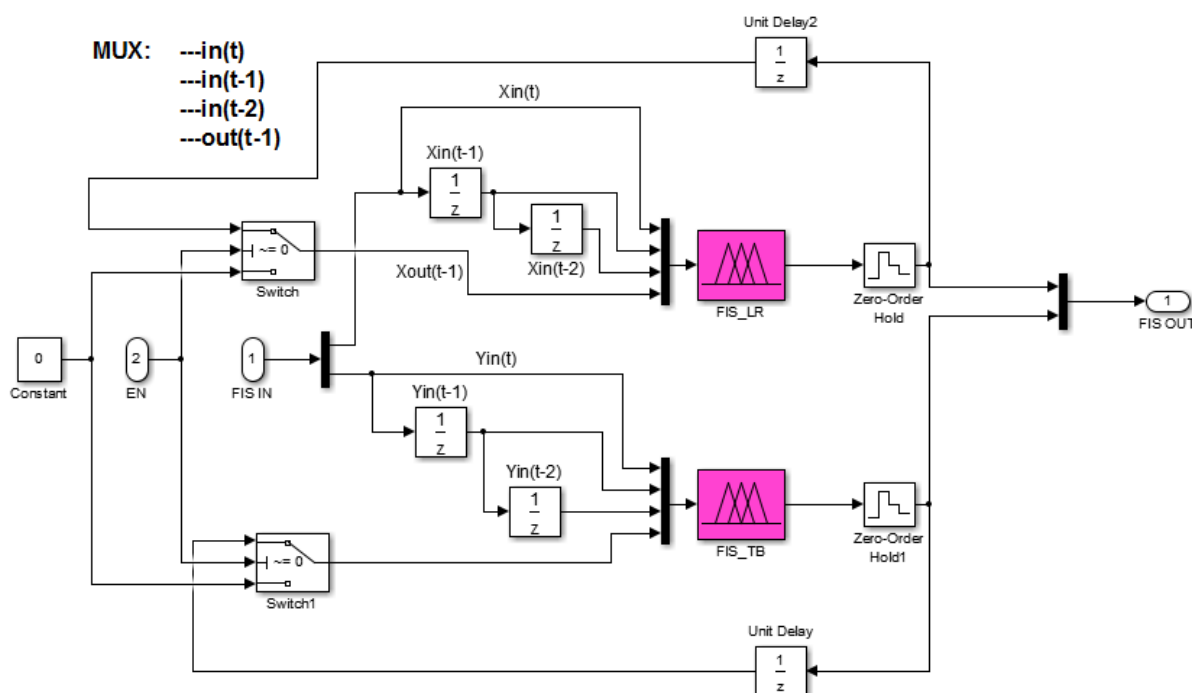
#### 4.4 Implementace regulátoru na bázi ANFIS a jeho zhodnocení

Obdobně jako při implementaci neuro-regulátoru, ani v tomto případě k regulaci modelu CE151 nestačí pouze naimportovat soubor s ANFISem do prostředí Simulink. Je nutné vytvořit vhodné schéma, které bude generovat vstupy, s kterými dokáže ANFIS nějakým způsobem pracovat. Pro lepší přehlednost se opět v rámci obecného schématu (viz Obrázek 45) vytvořil subsystém pouze pro ANFIS-regulátor. Schéma tohoto subsystému je znázorněno na obrázku níže (viz Obrázek 44).

Pomocné schéma je podobné jako u neuro-regulátoru. Protože jsou navržené regulátory systémy typu MISO, je však nutné oddělit řízení náklonu plošiny pro osu  $x$  a  $y$ . Blok „FIS\_LR“ řídí náklon doprava a doleva, tedy ovládá plošinu v ose  $x$ . Blok „FIS\_TB“ analogicky ovládá náklon plošiny nahoru a dolů, tedy v ose  $y$ . Tyto bloky propojují model v prostředí Simulink s externími „fis“ soubory, které vygeneroval nástroj Neuro-Fuzzy Designer.

Subsystém pro ANFIS-regulátor má dva vstupy a jeden výstup. Prvním vstupem je ovládání spuštění regulace po dokončení kalibrace. Tento vstup je označen jako „EN“. Druhý vstup s označením „FIS\_IN“ do subsystému přivádí regulační odchylku. Tento vstup má dvě složky. Jsou jimi regulační odchylka v ose  $x$  a regulační odchylka v ose  $y$ . Ze schématu je zřejmé, že tyto složky se téměř okamžitě po vstupu do subsystému dělí do dvou datových toků. Každý tento tok poté dělí na tři části. Signál dvou z nich se opozdí o  $T_S$  a  $2T_S$ , nebo v diskretní rovině o  $z^{-1}$  a  $z^{-2}$ . Stejně tak se opozdí i výstupy jednotlivých částí regulátoru o  $T_S$ , tedy  $z^{-1}$ . Tyto 4

signály se poté pomocí bloku „MUX“ pro každou část regulátoru spojují a jsou přiváděny na vstup těchto ANFISů. Výstup subsystému představuje dvě akční veličiny spojené do jednoho datového proudu pro ovládání náklonu plošiny v ose  $x$  a  $y$ .



**Obrázek 44:** Detailní schéma ANFIS-regulátoru

*Zdroj: vlastní*

Schéma je dále doplněno o popis veličin, které vstupují přes blok „MUX“ do ANFIS-regulátoru. Toto pořadí zapojení signálů je nutné dodržet, aby regulátor fungoval správně. Stejně tak je nutné dodržet pořadí zapojení samotných částí regulátoru po prvním rozdělení regulačních odchylek na dvě složky. První (shora) vždy představuje regulační odchylku v ose  $x$ , druhá v ose  $y$ . Tak jako u neuro-regulátoru je i toto schéma navíc doplněno o bloky pro odstranění případných neočekávaných chyb. Bloky „Zero-Order Hold“ drží konstantní hodnotu signálu v rámci jedné vzorkovací periody a nulová konstanta („Constant“) drží na vstupu regulátoru hodnotu 0 na místo opožděného výstupu, zatímco se provádí kalibrace. „Switch“ po kalibraci přepne uměně generování nuly na vstupu regulátoru na skutečné hodnoty opožděné akční veličiny.

Průměrné časy regulačního pochodu v sekundách jsou uvedeny v tabulce níže (viz Tabulka 5). Je zřejmé, že nově navržený regulátor na bázi CI dokáže spolehlivě nahradit původní PID regulátor. Časy regulačního pochodu jsou téměř stejné, pokud plošina vyrovnává ping-pongový míček, kterého se využilo i pro tvorbu trénovacích dat pro CI. Zajímavostí však je poměrně kratší doba regulace, pokud se pracuje s kovovým míčkem, který svými fyzikálními vlastnostmi připomíná nejbližší dokonalou kuličku.

**Tabulka 5:** Průměrné časy stabilizace RO pomocí ANFIS-regulátoru

<b>Typ použité kuličky</b>	<b>Čas stabilizace PID [s]</b>	<b>Čas stabilizace ANFIS-regulátoru [s]</b>
Ping-pongový míček	22,623	22,541
Kovová s pogumovaným povrchem	17,06	17,47
Kovová	18,06	16,265

*Zdroj: vlastní*

Tyto průměry se v případě ping-pongového míčku vypočítaly z padesáti provedených regulačních pochodů. Pro ověření adaptability se poté provedlo 10 měření dob regulace za použití kovové kuličky a kovové kuličky s pogumovaným povrchem. Dalšími analýzami těchto časů a regulačních pochodů se práce bude zabývat v kapitole 5.

## 5 POROVNÁNÍ NAVRŽENÝCH REGULÁTORŮ A JEJICH UMÍSTĚNÍ VE SCHÉMATU ŘÍDÍCÍHO SYSTÉMU

Tato část práce se bude zaměřovat na detailní zhodnocení výsledků této práce. V první části se popíše nově navržené schéma regulačního systému v prostředí Simulink. Poté se popíší průběhy regulace v závislosti na použitém typu kuličky, které plošina vyrovnává na určené místo. Tímto se zhodnotí adaptabilita nových regulátorů. Dále bude jedna podkapitola věnována analýze časů regulace, opět s rozdělením podle typů kuliček. Nakonec se i shrne složitost regulátorů na bázi CI a složitost jejich návrhu.

### 5.1 Nové schéma pro řízení modelu v prostředí MATLAB-Simulink

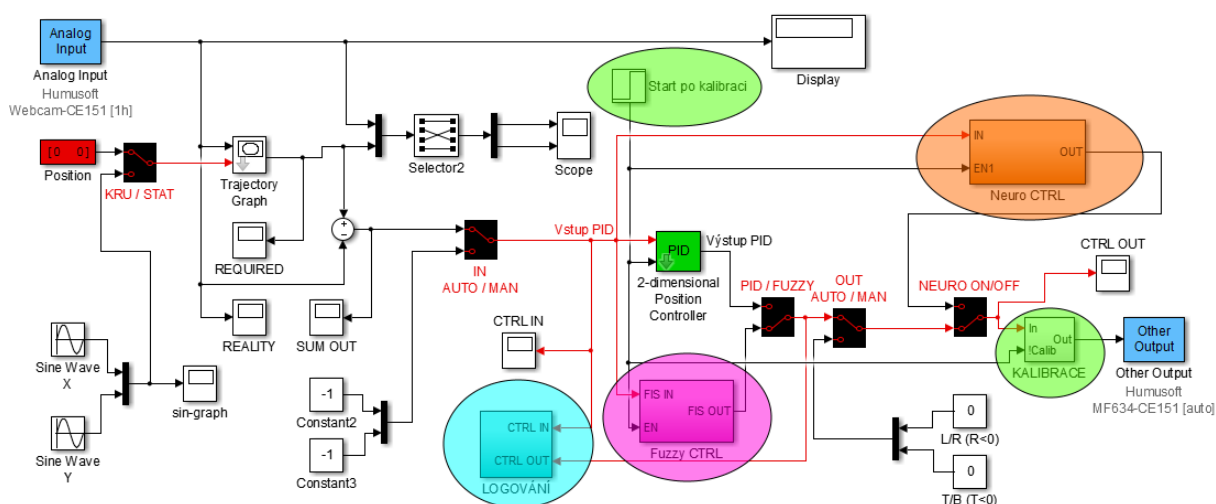
V úvodu práce bylo popsáno schéma regulačního systému (viz Obrázek 4), z kterého práce vycházela. Na obrázku níže (viz Obrázek 45) je uvedeno modifikované schéma pro řízení modelu CE151. Jsou v něm zakomponovány oba nové regulátory včetně dalších pomocných bloků, kterých se využívalo při návrhu. Hlavní změny částí schématu jsou odlišeny barvou. Modifikované schéma bez barevných odlišení a ve větším rozlišení je uvedeno v příloze A.

Mezi nově navržené pomocné bloky se řadí především umělé generování regulační odchylky na vstupu do regulátoru a stejně tak umělé generování akční veličiny na výstupu regulátoru. Těchto částí systému se využívalo hlavně při identifikaci chyb soustavy a porozumění funkce PID regulátoru. V implementacích prvních návrhů regulátorů na bázi CI se také využily pro generování statické regulační odchylky pro ověření jejich základního fungování. Dalšími objekty, které přímo nesouvisí s regulací soustavy jsou různé bloky typu „Scope“. Díky nim probíhala analýza různých částí systému v reálném čase. Pomohly také především identifikovat základní principy fungování regulačního systému, a to v reálném čase. Sledovala se většinou přechodová charakteristika a odpověď systému na různé podněty. Za zmínku také stojí všechny přepínače, kterými se nastavuje manuálně typ regulace ještě před spuštěním simulace. Díky nim není nutné vytvářet nové simulační modely pro každý typ regulace. Tyto pomocné bloky schématu blíže popisuje [3].

Jak již bylo zmíněno v kapitole 1.6.3, zeleně je označená část schématu, která představuje vylepšenou kalibraci soustavy a jednotné spuštění regulátorů. Spuštění regulátorů po kalibraci zajišťuje generátor jednotkového skoku v bloku „Start po kalibraci“, který je přiveden na vstup každého subsystému představujícího samotný regulátor.

Světle modrou barvou je označen subsystém pro ukládání dat z průběhu regulačního pochodu, tedy logování. Tento blok má na vstup přiveden signál regulační odchylky a akční veličiny.

Výstup nemá na schématu žádný, nicméně probíhá v něm export datového souboru s příponou „mat“ do pracovní paměti prostředí MATLAB. Blíže je tento blok popsán v kapitole 2.1.



**Obrázek 45:** Modifikované schéma pro řízení modelu CE151

*Zdroj: vlastní*

Hlavní inovaci, které se tato práce zabývá, představuje oranžově a růžově označená část schématu. Zde se nacházejí neuro-regulátor a ANFIS-regulátor. Jak je vidět, oba mají totožný vstup jako původní PID regulátor. Pomocí přepínačů „PID/FUZZY“ a „NEURO ON/OFF“ se dá nastavit požadovaný typ regulace. V průběhu regulačního pochodu však není vhodné tento typ měnit, dochází poté k destabilizaci systému a neočekávaným chybám.

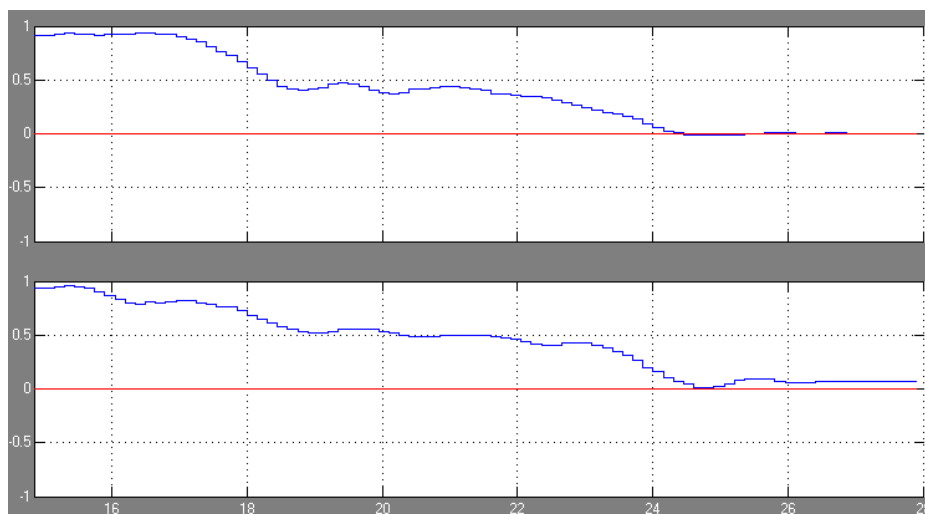
## 5.2 Průběhy regulace v závislosti na typu kuličky

V této kapitole se blíže popíší průběhy regulace včetně přibližných regulačních odchylek systému v ustáleném stavu. Všechny poznatky v následujících subkapitolách vycházejí z pozorování přes tří set regulačních pochodů. Každý regulátor vykazuje v rámci jednoho druhu kuličky při řízení soustavy přibližně stejné charakteristiky. Proto se také tato kapitola rozdělí do tří částí, přičemž v každé bude popsán průběh všech regulátorů.

### 5.2.1 Ping-pongový míček

Ping-pongový míček je původní ovládaný typ kuličky, na jehož základě se učila všechna použitá výpočetní inteligence. Typický průběh regulačního pochodu bez procesu kalibrace je již uveden v úvodních kapitolách (viz Obrázek 8), proto se na něj bude v této části práce pouze odkazovat. Jak již bylo řečeno, podle [13] se jedná o přechodovou charakteristiku zobrazující průběh regulačního pochodu s mírným přeregulováním (překmity). Oproti tomu si můžeme na obrázku níže (viz Obrázek 46 – osy a křivky vysvětleny v popisu Obrázku 8) povšimnout, že u nově navržených regulátorů se tyto překmity neobjevují. Jedná se tedy o přechodovou

charakteristiku s regulačním pochodem bez překmitů [13]. Tento průběh je typický pro oba typy regulátorů navržených na bázi CI. Tato skutečnost může mít později vliv na adaptabilitu regulátoru, pokud zkusíme plošinou vyrovnávat jiné typy kuliček.

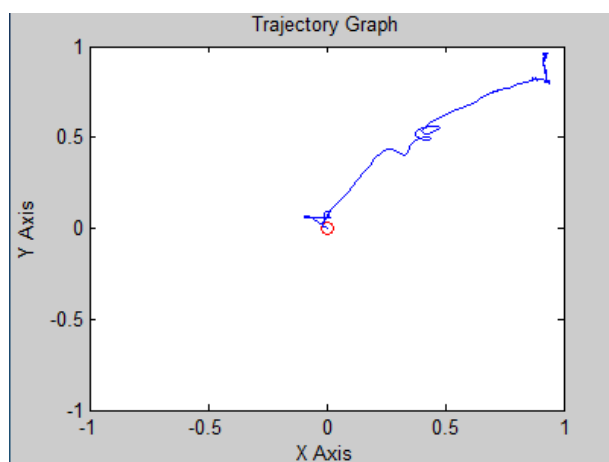


**Obrázek 46:** Přechodová charakteristika typického průběhu regulace regulátory CI

*Zdroj: vlastní*

Mírně odlišná vizualizace typického regulačního pochodu CI regulátorů je naznačena na obrázku níže (viz Obrázek 47). Jedná se o trajektorii míčku tak, jak se v průběhu stabilizace modelu CE151 pohyboval po náklonné plošině. Jak je vidět, míček se přibližoval středu, tedy cíli regulace, postupně oproti původní PID regulaci, kdy se plošina vychylovala o větší úhel a způsobila hlavně na počátku regulace větší překmit, jak kulička minula setrvačností cíl regulace. Ukázalo se však, že náklony plošiny o menší úhel při CI regulaci měly nevýhodu v tom, že se kulička s menší setrvačností občas zastavila o hranu svého povrchu. Tento problém je detailněji rozebrán v kapitole 1.5. Obecně se ale dá tvrdit, že CI velmi dobře generalizovala problém a nebyť nedokonalostí povrchu kuličky, průběh regulace by byl téměř dokonalý. Minimalizováním fyzikálních nedokonalostí kuliček se budou zabývat následující kapitoly.

S nedokonalostmi povrchu ping-pongového míčku také úzce souvisí velikost regulační odchylky v ustáleném stavu systému, tedy stavu, kdy již regulátor nevykazuje žádnou změnu akční veličiny a považuje soustavu za stabilizovanou. Čím blíže byla kulička cíli regulace, tím menší byl náklon plošiny a kulička se často zastavovala o svou hranu. Pokud se vyloučí extrémní případy, regulační odchylka v ustáleném stavu byla přibližně  $\pm 5\%$  a to jak pro původní PID regulátor, tak i regulátory na bázi CI. Extrémními případy se rozumí regulační pochody, které trvaly déle jak minutu právě kvůli problému popsanému v kapitole 1.5, kdy se kulička nepohybovala ani po zdatně nakloněné plošině.



**Obrázek 47:** Trajektorie kuličky pro typický průběh regulace ANFIS a neuro-regulátorem

*Zdroj: vlastní*

## 5.2.2 Kovová kulička s pogumovaným povrchem

Následující dva typy kuliček se použily pro ohodnocení adaptability regulátorů a jejich porovnání. První z nich byla kovová kulička s pogumovaným povrchem ze staršího modelu počítačové myši. Tato kulička má znatelně dokonalejší povrch bez hran, na kterých by se mohla zastavit při menším náklonu plošiny. Pogumovaný povrch by měl také zajistit menší setrvačnost.

Průběh regulace byl však překvapivě stejný pro všechny typy regulátorů. Časy regulace byly sice odlišné (viz Tabulka 4 a Tabulka 5), nicméně se vždy jednalo o regulační pochod s překmity. Velkou výhodou regulace této kuličky byla skutečnost, že nikdy nedošlo k extrémnímu případu, že by se kulička nedokázala ustálit na danou souřadnici ani po jedné minutě a déle. Regulační odchylka v ustáleném stavu byla menší než při regulaci s ping-pongovým míčkem, a to odhadem  $\pm 3\%$  od cíle regulace pro všechny typy regulátorů.

## 5.2.3 Kovová kulička

Kovové kuličky se využilo z toho důvodu, že má z již použitých nejdokonalejší fyzikální vlastnosti a od ping-pongového míčku se liší nejvíce. Váha kuličky je rozložena rovnoměrně v celém objemu a nedokonalosti povrchu jsou minimální. S tím souvisí i větší setrvačnost, tudíž se předpokládají vyšší nároky na regulátor, na které nebyl původně navržen. Může i nastat situace, kdy regulátor nebude schopen kuličku na plošině úspěšně vyrovnat na určenou souřadnici.

Následně provedené regulační pochody tyto předpoklady potvrdily. Bylo velmi zřetelně vidět, jak se systém s původním PID regulátorem dostává při některých měřeních na mez stability. Kulička poté nějakou dobu překmitávala v jedné ose náklonu plošiny z jednoho kraje na druhý,



a to až do doby, kdy se „vhodně“ odrazila od kraje na místo, kde regulátor dokázal správně zareagovat. Tento problém však nebyl tak znatelný u CI regulátorů. A právě zde se projevuje rozdíl v typických přechodových charakteristikách původní regulace a nové regulace při stabilizaci ping-pongového míčku. Vlivem většího překmitu PID regulátoru se oproti CI regulaci soustava mnohem snáze destabilizuje. Neuro-regulátor měl v některých regulačních pochodech podobný problém, nicméně netrvalo tak dlouho, aby se z ní dostal opět na postupnou stabilizaci systému. ANFIS-regulátor se překvapivě nedostal na mez stability ani jednou.

Závěrem můžeme tedy tvrdit, že podle pokusu s kovovou kuličkou jsou CI regulátory mnohem přizpůsobivější než PID, který byl navržen na vyrovnávání ping-pongového míčku. Přechodové charakteristiky všech regulátorů vykazovaly překmit, nicméně u PID regulátoru byl značně větší, než došlo ke stabilizaci. Regulační odchylka v ustáleném stavu byla díky hladkému povrchu kovové kuličky ještě menší než při řízení kovové kuličky s pogumovaným povrchem, přibližně +/- 1 %.

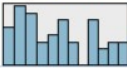
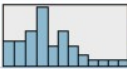
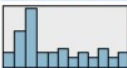
### 5.3 Porovnání průměrných časů stabilizace

Snad nejdůležitější charakteristikou je v rámci ohodnocení nově navržených regulátorů na bázi CI doba regulace. Toto hodnocení se zakládá na padesáti nezávisle provedených regulačních pochodech za použití ping-pongového míčku a deseti pro případ kovové kuličky s pogumovaným povrchem a pouze kovové. Tyto počty měření se provedly pro každý z regulátorů – původní PID, ANFIS a neuro. Celkem se tedy analýza zakládá na 210 průbězích regulace. Vyšší počet měření pro ping-pongový míček se prováděl z toho důvodu, že nerovný povrch kuličky značně ovlivňoval výsledky měření. Provedlo se jich tedy více, aby tato chyba ve výsledných datech byla vlivem průměrování výsledných časů co možná nejmenší. Aby však úplně nezanikla a aby existovala veličina, která bude zachycovat vliv fyzikálních nedokonalostí kuliček na regulaci, využije se i směrodatné odchylky změřených časů. Všechny tyto pohledy na data se opět prováděly v IBM SPSS Modeler vzhledem k tomu, že jejich vizualizace je v tomto prostředí velice snadná.

Následující tabulky představují základní statistiky provedených měření dle typu kuličky. První sloupec popisuje typ použitého regulátoru. Druhý sloupec, „Graph“, je zjednodušený histogram všech změřených dob regulačních pochodů. „Measurement“ označuje typ veličiny, pro kterou se provádí následné výpočty. V tomto případě čas je vždy spojitý. „Min“ a „Max“, jak názvy napovídají, představují minimální a maximální dobu regulace, stejně tak „Mean“ udává průměr. „Std. Dev“ z anglického standard deviation značí směrodatnou odchylku. Sloupec „Skewness“ dává informaci o koeficientu šikmosti, nicméně tuto veličinu práce nebude dále komentovat,

jelikož nepodává žádnou hodnotnou informaci. „Unique“ se používá pouze u nespojitých veličin pro vyjádření počtu odlišných hodnot. Poslední sloupec s označením „Valid“ udává, z kolika záznamů se statistiky počítaly.

Obrázek 48 udává základní statistiky pro padesát časů regulačních pochodů v rámci jednotlivých typů použitých regulátorů. Minimální a maximální časy regulace jsou velmi podobné, nicméně první odlišnosti jsou viditelné už na první pohled v histogramu. Ideální stav by nastal, pokud by časy regulace měly normální rozdělení pravděpodobností. Znamenalo by to, že regulátor při každém regulačním pochodu řídí soustavu na základě stejných závislostí mezi regulační odchylkou a akční veličinou. Doby regulace by poté oscilovaly blízko průměrné době regulace. Odlišnější hodnoty by se objevovaly pouze zřídka v případě, kdy například regulátor začíná řídit soustavu s menší regulační odchylkou. Toto chování však model CE151 nevykazuje a nejlepším vysvětlením jsou nedokonalosti na povrchu kuličky. Pokud se kulička zastaví o hranu svého povrchu, regulátor postupně naklání plošinu, dokud se kulička neuvolní, což ale způsobí díky její náhlé větší setrvačnosti větší destabilizaci systému a tím pádem delší dobu stabilizace. Vlivy těchto nedokonalostí spadají pod tzv. chybové veličiny, které běžně v praxi působí na řízenou soustavu. Míru toho, jak chybová veličina ovlivňuje řízení systému tedy poměrně dobře vystihuje směrodatná odchylka dob regulace.

Field	Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
PID		Continuous	10.400	39.650	22.623	8.585	0.461	--	50
anfis		Continuous	11.900	42.800	22.541	6.733	0.903	--	50
neuro		Continuous	11.300	39.950	22.577	8.228	0.700	--	50

**Obrázek 48:** Charakteristiky dob regulace ping-pongového míčku

*Zdroj: vlastní*

Na první pohled se tedy zdá, že výpočetní inteligence velmi dobře napodobuje původní PID regulátor a v ničem nevyčnívá. Průměrné doby regulace jsou téměř identické. Nicméně pokud do porovnávání zahrneme i směrodatné odchylky, vidíme drobný rozdíl mezi PID a neuro-regulátorem a oproti nim velkou odchylku od ANFIS-regulátoru. Znamená to tedy, že systém řízený ANFIS-regulátorem není tolik ovlivněn chybovou veličinou. Vzhledem k počtu naměřených časů se ani nedá tvrdit, že by se jednalo o náhodu. Rozdíly směrodatných odchylek jednotlivých typů regulace dokazuje i zjednodušený histogram. V případě PID regulace je zřejmá značná nahodilost mezi změřenými časy.

Dále se provedlo identické měření dob regulace pro kovovou kuličku s pogumovaným povrchem. Počet měření byl nižší z důvodu omezení chybové veličiny vlivem lepších

fyzikálních vlastností použité kuličky. Analýza těchto časů je uvedena v tabulce níže (viz Obrázek 49). Vzhledem k menšímu počtu měření nemají maximální a minimální doby regulace takovou vypovídací schopnost, takže je práce nebude komentovat ani v případě kovové kuličky. V případě kovové kuličky s pogumovaným povrchem jasně vyniká regulace samotnou neuronovou sítí. Nejen že průměrná doba regulace je přibližně o 20 % menší, ale i směrodatná odchylka naznačuje, že je řízení velmi málo zatíženo chybovou veličinou.

Field	Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
PID		Continuous	9.650	25.700	17.060	4.057	0.480	--	10
anfis		Continuous	13.250	27.500	17.470	4.178	1.680	--	10
neuro		Continuous	11.900	17.900	13.915	1.702	1.496	--	10

**Obrázek 49:** Charakteristiky dob regulace kovové kuličky s pogumovaným povrchem

*Zdroj: vlastní*

Jako poslední se provedla analýza časů regulačního pochodu pro kovovou kuličku, která je znatelně menší než ping-pongový míček a má také fyzikální vlastnosti, které se nejvíce blíží těm ideálním. V tabulce níže (viz Obrázek 50) jsou uvedeny charakteristiky deseti měření v rámci všech regulátorů v případě řízení soustavy s touto kuličkou. Tyto charakteristiky ukazují, že nejlepší schopnost adaptability má v tomto případě ANFIS-regulátor, který má průměrnou dobu regulace o 10 % menší než původní PID regulátor. Řízení soustavy ANFIS-regulátorem také vykazuje nejnižší směrodatnou odchylku, která je v případě PID regulátoru velmi vysoká kvůli problémům s jeho adaptabilitou na řízení jiného typu kuličky. Kvůli největším rozdílům ve fyzikálních vlastnostech použitých míčků se dá tvrdit, že ANFIS je tedy nejprizpůsobivější ze všech použitých typů regulátorů.

Field	Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
PID		Continuous	7.700	35.900	18.060	10.262	0.709	--	10
anfis		Continuous	12.500	18.950	16.265	2.135	-0.604	--	10
neuro		Continuous	13.550	20.150	17.510	2.566	-0.500	--	10

**Obrázek 50:** Charakteristiky dob regulace kovové kuličky

*Zdroj: vlastní*

Pro celkové porovnání původní regulace a regulace na bázi výpočetní inteligence můžeme využít průměry dílčích průměrných dob regulace a průměry směrodatných odchylek všech výše provedených analýz v rámci jednotlivých regulátorů. Tyto průměry jsou uvedeny v tabulce níže (viz Tabulka 6).

**Tabulka 6:** Srovnání průměrných směrodatných odchylek všech regulačních pochodů

	<b>PID</b>	<b>ANFIS</b>	<b>Neuro</b>
<b>Průměr průměrných dob regulace [s]</b>	19,25	18,76	18,00
<b>Průměrné směrodatné odchytky</b>	7,63	4,35	4,17

*Zdroj: vlastní*

Pokud se zaměříme pouze na průměry časů, lze tvrdit, že výpočetní inteligence věrně napodobuje původní PID regulátor. Časy jsou si velmi podobné. Toto však nelze tvrdit o průměrných směrodatných odchytkách. Ty jsou u původní regulace téměř dvojnásobné. Tato skutečnost vypovídá o mnohem lepší schopnosti adaptability CI. CI tedy dokázalo nejen napodobit původní regulaci, ale ještě více zobecnit celý regulační problém. Toto tvrzení je také podloženo velmi podobnou průměrnou směrodatnou odchylkou obou CI regulátorů. Nejedná se tedy o náhodu, ale metody CI musely dojít k podobným závěrům týkajících se řízení modelu CE151. Při bližším zkoumání ANFIS-regulátoru by bylo možné tyto závěry do jisté míry identifikovat, nicméně to už není předmětem této práce.

#### **5.4 Náročnost návrhů řídicích systémů**

Typy regulátorů, které jsou použity v této práci lze porovnat i z hlediska jejich samotných návrhů. Tato kapitola se zaměří právě na to. Při posuzování návrhu původního PID regulátoru se bude vycházet z [1]. Protože vždy při návrhu regulátoru záleží na úrovni znalostí experta, níže popsané porovnání se spíše zaměří na návrh z obecnějšího úhlu pohledu a bude předpokládat pouze základní znalosti navrhovatele.

V případě návrhu PID regulátoru se expert nemusí příliš zabývat jeho vnitřní strukturou. Složky P, I a D jsou již definovány a jejich samotná funkce popsána. Problémem je však vhodné nastavení parametrů, aby regulační proces probíhal co nejefektivněji. Této problematice musí expert rozumět. S tím souvisí také pokročilejší znalost matematiky nebo nástrojů, které se optimalizací PID regulátoru zabývají.

V případě návrhu regulátorů na bázi výpočetní inteligence nejsou požadavky na znalost matematických postupů tak velké. Velkou výhodou oproti návrhu PID je také skutečnost, že postup návrhu je možno aplikovat u jakéhokoli regulačního problému, a to bez hlubší znalosti celého systému. Pokud jsou známy kombinace vstupů a výstupů, stačí CI regulátor navrhnout s určitým počtem opožděných regulačních odchylek a akčních veličin na vstupu, naučit samotnou metodu CI a naimplementovat ji. Existují dokonce i metody, kdy se CI učí sama podle regulační odchylky, nicméně těmi se tato práce nezabývá.

Pokud porovnáme návrhy ANFIS a neuro-regulátoru, oba jsou podobně náročné, nicméně u neuronové sítě nebylo nutné nastavovat tolik parametrů. Prvním společným krokem bylo

zvolení struktury samotné metody, tedy počet vstupů a výstupů. Tvorba trénovacího souboru dat byla o trochu složitější u ANFIS-regulátoru. Musely se vytvořit celkem 4 soubory, a to pro trénovací a testovací množinu dat pro každou osu řízení modelu zvlášť. Neuronové sítě stačily pouze 2 soubory a rozdělení na trénovací a testovací množinu proběhlo v samotném nástroji prostředí MATLAB. U NS bylo nutné nastavit poměr mezi trénovací, testovací a validační množinou dat a počet neuronů ve skryté vrstvě. ANFIS vyžadoval složitější konfiguraci návrhu, nicméně toto může být vnímáno i jako výhoda. Nabízely se kupříkladu různé tvary a počty funkcí příslušnosti, výběr mezi dvěma algoritmy učení a počty epoch učení. I přes to však nároky na pochopení všech aspektů návrhu nejsou tak vysoké jako při návrhu a optimalizaci PID regulátoru.

## 5.5 Složitost struktury regulátorů

Pokud porovnáme detaily schémat PID (viz Obrázek 7), ANFIS (viz Obrázek 44) a neuro-regulátoru (viz Obrázek 35), mohlo by se zdát, že jsou všechny podobně složité. Není tomu ale tak. V případě regulátorů na bázi CI je v jejich schématech ještě jeden, či více skrytých subsystémů. Obecně vzato, CI je navrženo na řešení velmi složitých problémů reálného světa a tomu také odpovídají nároky na výpočetní výkon. PID regulátor pouze provádí operace s veličinami, pro které byl navržen. Regulátory na bázi výpočetní inteligence jsou tedy mnohem složitější.

Pokud porovnáme ANFIS a neuro-regulátor, složitější strukturu má ANFIS. Toto plyne už z jeho názvu – jedná se o spojení neuronové sítě a fuzzy logiky. ANFIS je v podstatě neuronovou sítí, která se chová jako systém na bázi fuzzy logiky. Pokud porovnáme tyto dvě navržené NS v rámci této práce (viz Obrázek 34 a Obrázek 43), ANFIS obsahuje podstatně více neuronů, tudíž i jeho nároky na výpočetní výkon budou vyšší.

Z hlediska vstupů a výstupů samotných regulátorů má PID velkou výhodu. Obsahuje pouze dva vstupy a dva výstupy. Přesněji jde o jeden vstup a jeden výstup pro ovládání každé osy náklonu plošiny. Z testování trénovacích dat v kapitole 2.3 však u regulátorů na bázi CI vyplynulo, že je nutné navrhnout 8 vstupů do regulátoru a 2 výstupy. A toto bylo řešení, které počet vstupů minimalizovalo, jinak by CI regulátor mohl mít vstupů i více. V případě řízení modelu CE151 má tedy PID regulátor nejméně složitou strukturu. Pokud by však systém vyžadoval konstrukci regulátoru s více vstupy a jedním výstupem, PID by se implementoval velmi těžko. Jak je totiž vidět na detailním schématu PID regulátoru (viz Obrázek 7), jedná se o SISO systém a každou složku regulační odchylky řídí jeden samostatný PID regulátor.

## ZÁVĚR

Cílem diplomové práce bylo navrhnout nové způsoby řízení výukového modelu CE151, které budou pracovat na bázi metod CI. Toho se úspěšně docílilo a nyní práce shrne jednotlivé milníky celého projektu.

První oblastí, kterou se práce zabývala, byl popis výukového modelu se zaměřením na proces regulace. Po stručném charakterizování soustavy, tedy modelu CE151, se detailně popsal původní PID regulátor. Následovala identifikace problémů s původní regulací. Hlavními problémy bylo prokluzování lanek, která naklánějí plošinu soustavy, zkreslování souřadnic míčku při snímání plošiny, značné nedokonalosti povrchu původní kuličky a různé hodnoty akční veličiny při dokončení regulace. Všechny tyto problémy práce více či méně zmírnila natolik, aby následný návrh CI regulátorů probíhal bez zbytečných komplikací.

Po zlepšení regulačního procesu následovalo získávání dat pro učení metod CI. V prostředí MATLAB-Simulink se vytvořil systém, který zaznamenával vstupy a výstupy PID regulátoru, a to i s opožděními o několik vzorkovacích period. Poté se provedlo několik logovaných regulací a výstupem bylo 20 datových souborů. Jejich kompletace, úprava, testování a příprava pro nástroje návrhu CI byla provedena v prostředí IBM SPSS Modeler.

Pomocí teorie PID regulátorů a ověřováním získaných trénovacích dat pomocí neuronové sítě se rozhodlo o struktuře CI regulátorů. Pomocí nástrojů Neural Network Fitting App a Neuro-Fuzzy Designer v rámci prostředí MATLAB se zkonstruovaly systémy výpočetní inteligence, které s vysokou přesností napodobují chování PID regulátoru. Jednalo se o neuronovou síť a adaptivní neuro-fuzzy inferenční systém.

Nově navržené systémy CI (neuro-regulátor a ANFIS-regulátor) se následně implementovaly do schématu regulačního systému v prostředí MATLAB-Simulink. Stabilizováním soustavy při minimální regulační odchylce se po spuštění regulace potvrdila funkčnost CI regulátorů a dobou regulace se také prokázala vhodnost řešení oproti původnímu způsobu regulace.

Na závěr se provedlo několik desítek měření dob regulace pro všechny typy regulátorů při použití nejen původní kuličky, ale také dvou dalších s jinými fyzikálními vlastnostmi. Sledoval se navíc i přibližný průměr regulační odchylky v ustáleném stavu systému a samotný průběh regulace.

Závěrečné analýzy výsledků testování CI regulátorů jasně poukazují na úspěšné nahrazení původního způsobu regulace s mnohem lepší schopností adaptability na úkor lehce vyšších nároků na výpočetní výkon, což je ale v této době přijatelné.

## POUŽITÁ LITERATURA

- [1] BALÁTĚ, Jaroslav. *Automatické řízení*. 2., přeprac. vyd. Praha: BEN - technická literatura, 2004. 663 s. ISBN 80-7300-148-9.
- [2] HAYKIN, Simon S. *Neural networks: a comprehensive foundation*. 2nd ed. Upper Saddle River: Prentice Hall, ©1999. 842 pages. ISBN 0-13-273350-1.
- [3] HEGR, Jindřich. *Návrh příkladů pro výukový model systému řízení*. Pardubice, 2016. 45 stran. Bakalářská práce na Fakultě ekonomicko-správní Univerzity Pardubice na Ústavu systémového inženýrství a informatiky.
- [4] HORÁČEK, Petr. *CE 151 BALL & PLATE APPARATUS: Educational Manual*. Revision 3.2. Prague: Czech Technical University of Prague, 2008. 33 s.
- [5] JANG, Jyh-Shing R. – SUN, Chuen-Tsai – MIZUTANI, Eiji. *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. 1st ed. Upper Saddle River: Prentice-Hall, ©1997. 614 pages. ISBN 0-13-261066-3.
- [6] JURA, Pavel. *Základy fuzzy logiky pro řízení a modelování*. 1. vyd. Brno: VUTIUM, 2003. 132 s. ISBN 80-214-2261-0.
- [7] KARBAN, Pavel. *Výpočty a simulace v programech Matlab a Simulink*. 1. vyd. Brno: Computer Press, 2006. 220 s. ISBN 80-251-1301-9.
- [8] KIRCHNER, Miroslav. *Laboratorní model CE 151 – Kulička na ploše*. Zlín, 2011. 79 stran. Diplomová práce na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně na Ústavu řízení procesů.
- [9] NGUYEN, Hung T. – WALKER, Elbert A. *A first course in fuzzy logic*. 3rd ed. Boca Raton: Chapman & Hall, 2006. 440 pages. ISBN 1-58488-526-2.
- [10] NILSSON, Nils J. *Artificial Intelligence: A New Synthesis*. 1st edition. Morgan Kaufmann Publishers, 1998. 513 pages. ISBN 7-111-07438-6.
- [11] POKORNÝ, Miroslav. *Umělá inteligence v modelování a řízení*. 1. vyd. BEN - technická literatura, 1996. 187 s. ISBN 80-901984-4-9.
- [12] RUSSEL, Stuart – NORVIG, Peter. *Artificial Intelligence: A Modern Approach*. 3rd edition. Upper Saddle River, New Jersey 07458: Pearson Education, 2010. 1152 pages. ISBN 0-13-604259-7.

- [13] VORÁČEK, Rudolf – ANDRÝSEK, František – BRÝDL, Zdeněk – KOHOUT, Luděk – ŠMEJKAL, Ladislav. *Automatizace a automatizační technika II*. 1.vyd. Praha: Computer Press, 2000. 218 s. ISBN 80-7226-247-5.



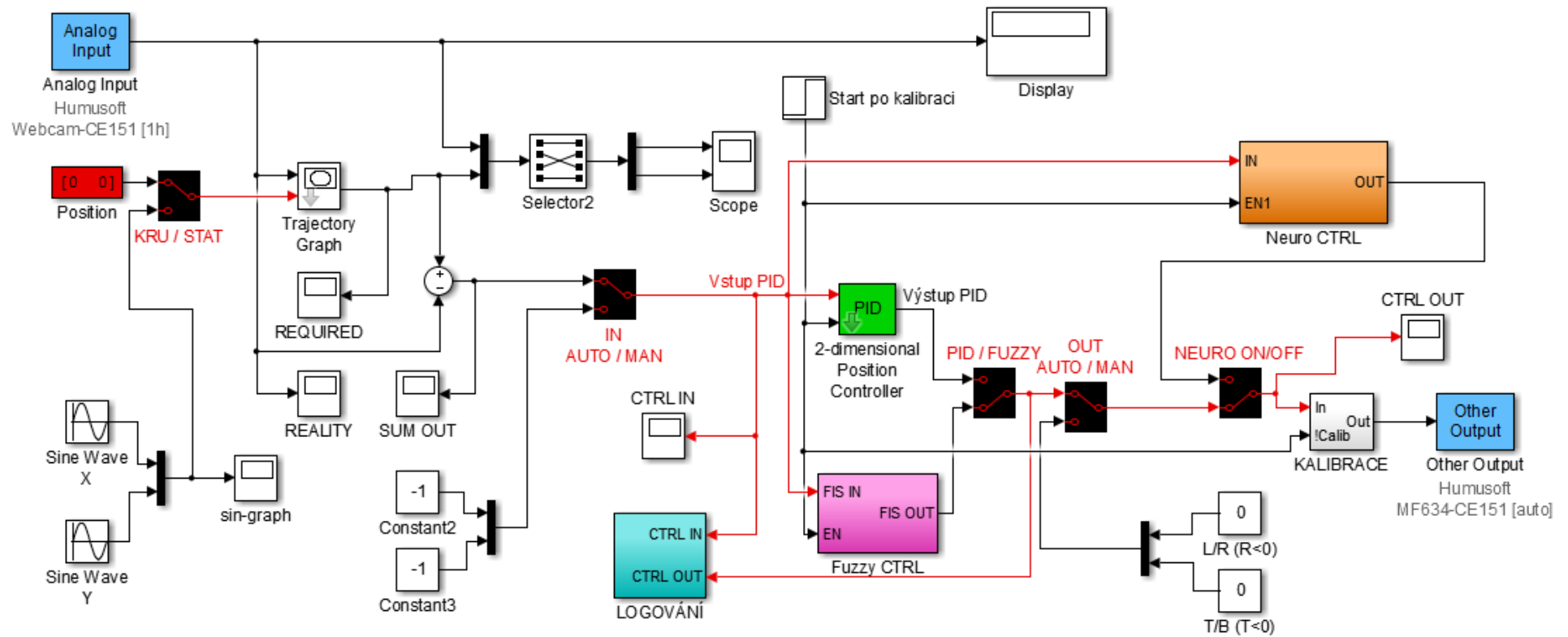
## **SEZNAM PŘÍLOH**

Příloha A – Schéma modifikovaného řídicího systému

Příloha B – Schéma kompletace datových souborů v IBM SPSS Modeler

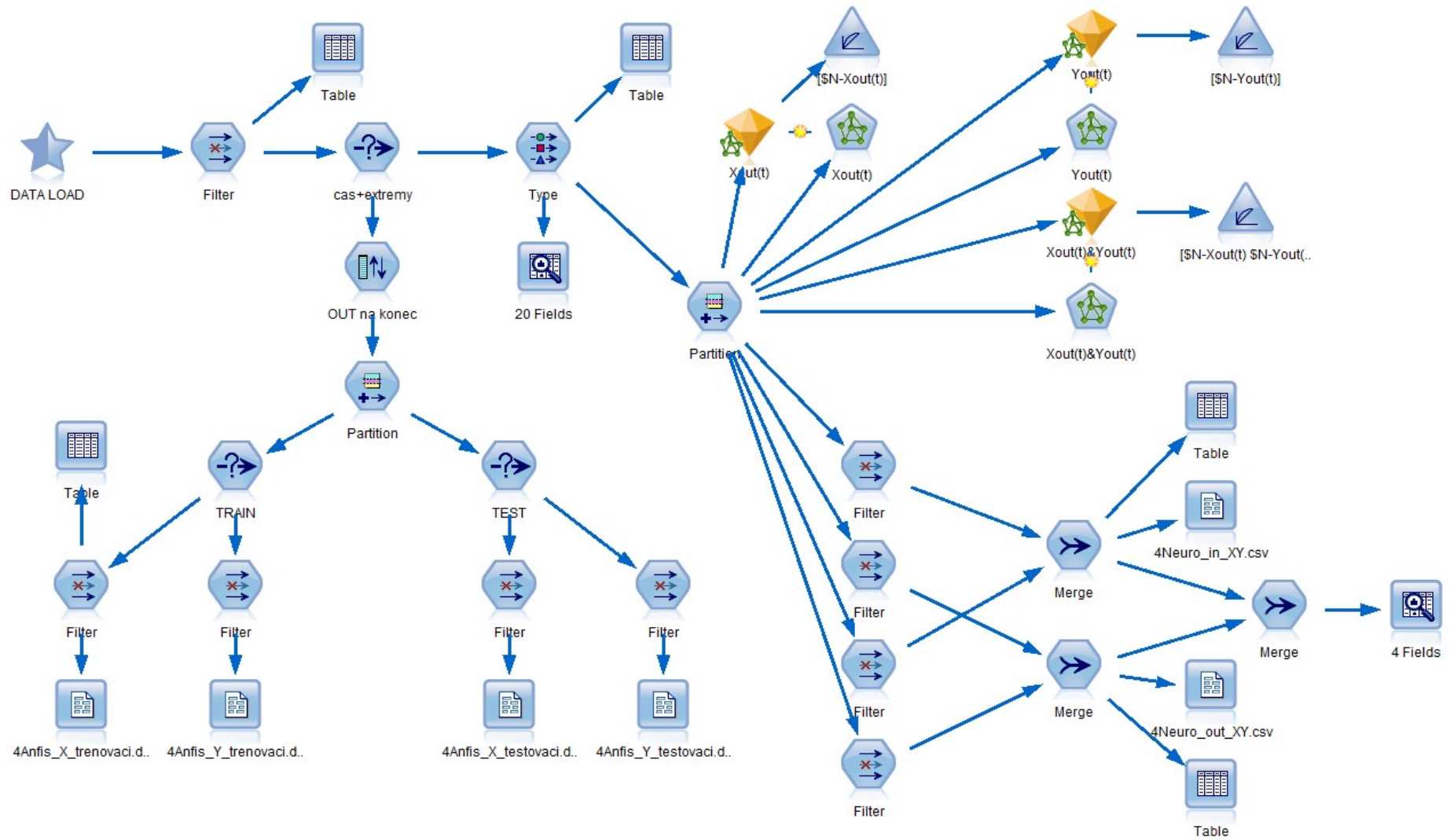
Příloha C – Detailní schéma původního PID regulátoru

# PŘÍLOHA A – SCHÉMA MODIFIKOVANÉHO ŘÍDÍCÍHO SYSTÉMU



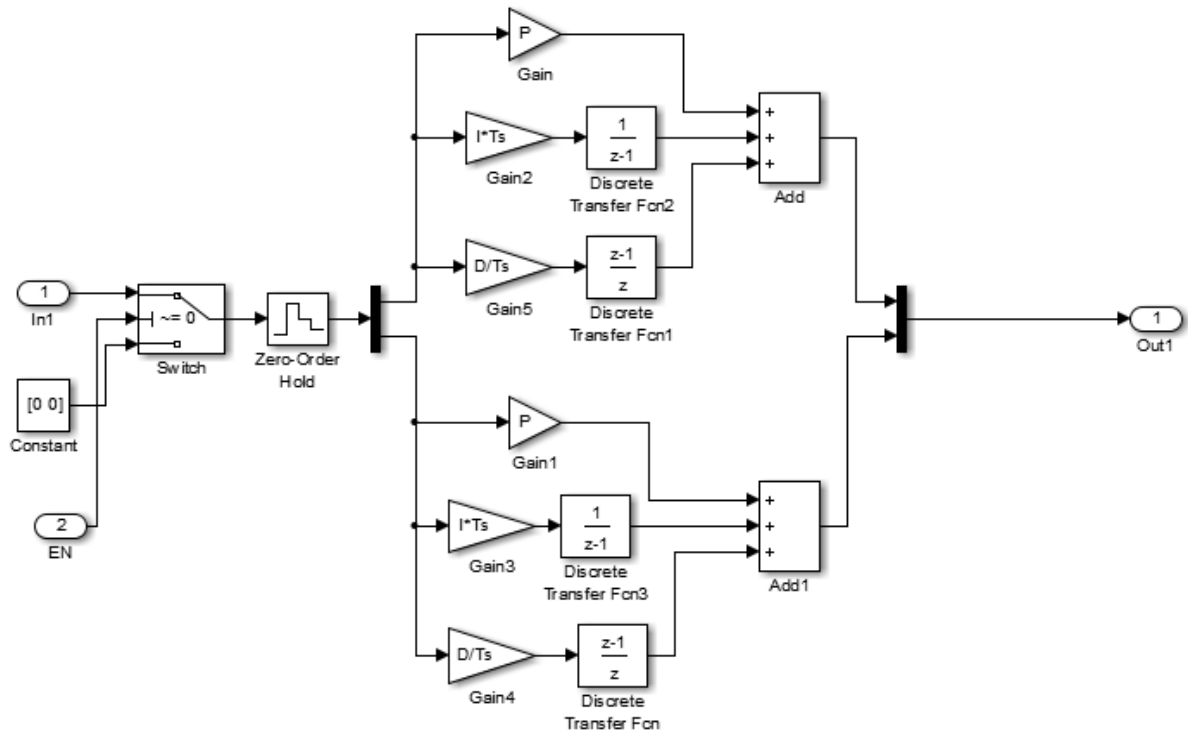
Zdroj: vlastní

## PŘÍLOHA B – SCHÉMA KOMPLETACE DATOVÝCH SOUBORŮ V IBM SPSS MODELER



Zdroj: vlastní

## PŘÍLOHA C – DETAILNÍ SCHEMA PŮVODNÍHO PID REGULÁTORU



Zdroj: převzato z mdl-souboru k [4]