

Metódy soft computing v priemyselných riadiacich systémoch

Ing. Ladislav Körösi, PhD.
ÚRK, FEI STU

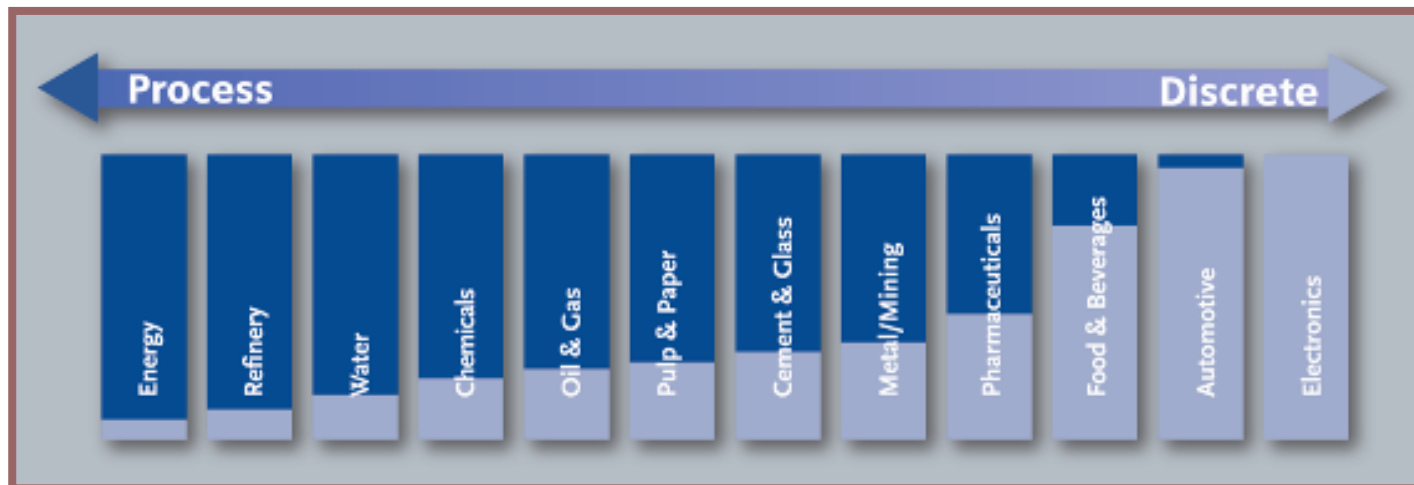
ladislav.korosi@stuba.sk

Automatizácia v priemysle

Tok produkcie (výroby)



Automatizácie pre niektoré odvetvia



Programovateľný logický automat

Pojem PLC

- „Programovateľný logický automat“
 - anglicky „Programmable logic controller (PLC)“
 - nemecky „Speicherprogrammierbare Steuerung (SPS)“

Užívateľsky programovateľný číslicový počítač, ktorý má oproti bežným počítačom niektoré špecifické vlastnosti (odolnosť voči zvýšenej teplote, vibráciám, vlhkosti, ...).

Programovateľný logický automat

Pojem PAC

- Programmable automation controller (PAC)

Pojem, ktorý sa používa na opis ľubovoľného typu riadiaceho systému, ktorý obsahuje pokročilé inštrukcie (operácie s vektormi, maticami, rôzne štruktúry PID regulátorov, fuzzy či neuro systémy a pod.). Z pohľadu funkcionality je PAC programovateľný logický automat s niektorými vlastnosťami (a inštrukciami) DCS.

Programovateľný logický automat

Programový cyklus PLC

Režimy PLC:

- Run – Klasický stav. Uložený program sa vykonáva.
- Stop – Vykonávanie programu bolo zastavené užívateľom alebo blokujúcou chybou programu alebo z iného dôvodu

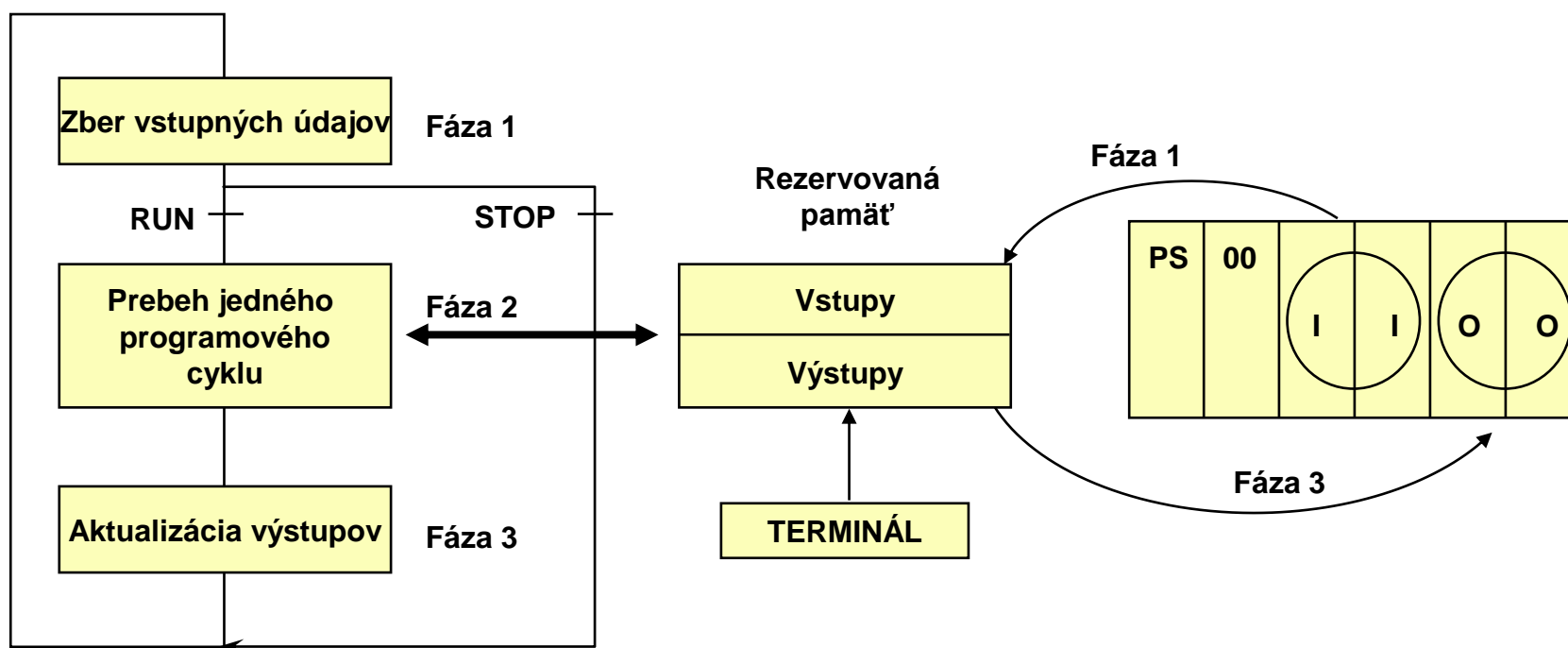
Spôsoby prepnutia režimov:

- Softvérovo (len z RUN -> STOP; napr. S7-1200)
- Prepínačom na CPU (napr. S7-1500)

Programovateľný logický automat

Programový cyklus PLC

Zjednodušený programový cyklus PLC



Programovateľný logický automat

Programový cyklus PLC

- 1. fáza: Procesor „zmapuje“ stav logiky vstupov a získaný obraz prenáša do dátovej pamäte (process image vstupov)
- 2. fáza: Postupné vykonávanie (logických) operácií uložených v pamäti, pričom sa používa obraz stavu vstupov uložených v dátovej pamäti. Výsledky každej logickej operácie sa prenášajú do dátovej pamäti (obraz výstupov – process image výstupov)
- 3. fáza - Prekopíruje stavy logických obrazov výstupov (uložených v dátovej pamäti) do modulov výstupov

Programovateľný logický automat

Programový cyklus PLC

- Dĺžka cyklu - Čas cyklu (alebo vykonania) je čas medzi dvoma operáciami načítania vstupných údajov. Zahŕňa 3 fázy a systémové operácie (diagnostika, komunikácia, ...)
- Ochranný čas (watchdog) – Čas po ktorom sa PLC prepne do stavu STOP (nedokončenie cyklu).

Programovateľný logický automat

Programový cyklus PLC

Programový cyklus (tzv. úlohy - tasks):

- Cyklický – Po dokončení fázy 3 sa hneď pokračuje fázou 1 bez čakania
- Periodický – Fázy 1 až 3 sú spúšťané v presne definovaný časový interval
- Udalostný – Fázy 1 až 3 sa vykonajú pri príchode udalosti (napr. stavu digitálneho vstupu alebo časovania)

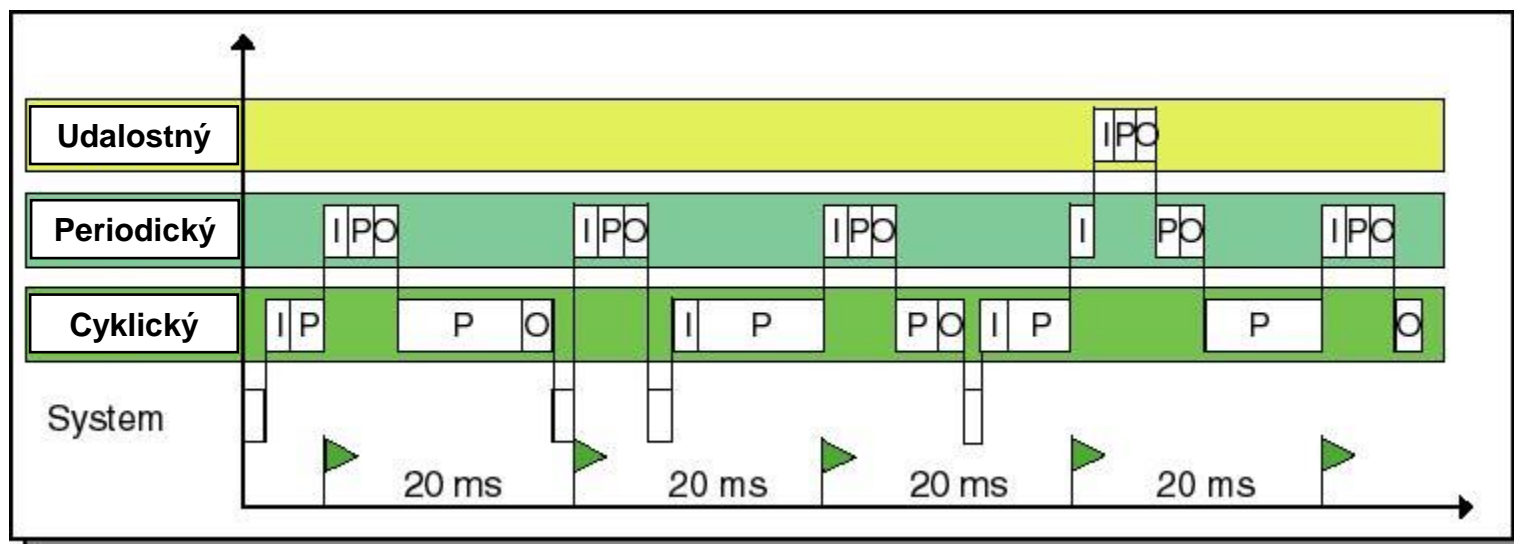
Úlohy sa v niektorých softvéroch volajú ako organizačné bloky.

Programovatelný logický automat

Programový cyklus PLC – „Multitasking“

Príklad viacerých úloh (Schneider el.)

Priority pevne stanovené: Cyklická úloha (najnižšia),
Udalostná úloha (najvyššia)



Programovatelný logický automat

Výrobci

Mitsubishi

Rockwell Automation

Schneider Electric

Siemens

Programovateľný logický automat

Programovacie jazyky

Jazyky (Norma IEC 61131-3):

- Grafické
 - Rebríkový diagram (angl. Ladder diagram - LD)
 - Diagram funkčných blokov (angl. Function Block Diagram - FBD)
 - Sekvenčný diagram (angl. Sequential function chart - SFC)
- Textové
 - Štruktúrovaný text (angl. Structured text - ST)
 - Zoznam inštrukcií (angl. Instruction list- IL, Statement list - STL)

Iné...

Programovateľný logický automat

Programovacie jazyky – ST (príklady)

```
%M10:=%M0 AND(%M1 OR NOT %M2) AND %M3 AND %M4;  
IF %M0 AND(%M1 OR NOT %M2)AND %M3 AND NOT %M4 THEN  
    SET %M11;  
END_IF;
```

```
if not x.na_snimaci and x.hist then  
    if x.pohyb_plus then x.aktualna_poloha:=x.aktualna_poloha+0.5; end_if;  
    if x.pohyb_minus then x.aktualna_poloha:=x.aktualna_poloha-0.5; end_if;  
end_if;
```

```
FOR premenna:=1 TO hodnota DO  
    prikaz1; prikaz2;  
    // EXIT na predcasne ukoncenie FOR  
END_FOR;
```

Programovateľný logický automat

Premenné – Elementárne dátové typy a UDT

Dátové typy premenných (tagov) môžeme rozdeliť na:

- Elementárne dátové typy
 - INT
 - DINT
 - REAL
 - ...
- Používateľom definované dátové typy (UDT – user defined data types v Siemens RS, DDT – derived data types v Schneider EI. RS)

Programovateľný logický automat

Premenné – Elementárne dátové typy a UDT (Príklad z Unity Pro)

Používateľom definované dátové typy (tzv. štruktúry) sa skladajú z elementárnych dátových typov. Pre menšie a staršie PLC nie sú podporované. Vo väčšine prípadov umožňujú viacnásobne vnorené štruktúry použiteľné aj ako (viacrozmerné) pole (ARRAY). Príklady v Unity Pro:

Užívateľom def. premenné:

Štruktúra:

Name	Type
str_nieco	<Struct>
prva	BOOL
poc	INT
ine	REAL

Name	Type
prem_1	str_nieco
prva	BOOL
poc	INT
ine	REAL
prem_2	ARRAY[1..2] OF str_nieco
prem_2[1]	str_nieco
prva	BOOL
poc	INT
ine	REAL
prem_2[2]	str_nieco

Programovateľný logický automat

Premenné – Elementárne dátové typy a UDT (Príklad z Unity Pro)

Adresácia pomocou bodkovej syntaxe,
(t.j. NAZOV.VNORENE1.VNORENE2.ATD)

Príklady z obr.:

prem_1.prva

prem_2[1].prva

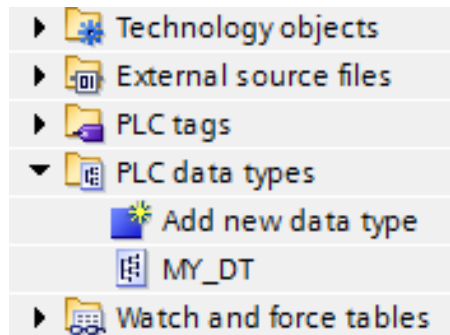
prem_2[2].prva

Name	Type
[-] prem_1	str_nieco
[-] prva	BOOL
[-] poc	INT
[-] ine	REAL
[-] prem_2	ARRAY[1..2] OF str_nieco
[-] prem_2[1]	str_nieco
[-] prva	BOOL
[-] poc	INT
[-] ine	REAL
+ prem_2[2]	str_nieco

Programovateľný logický automat

Premenné – Elementárne dátové typy a UDT (Príklad z TIA Portal)

TIA Portal – UDT v dátovom bloku (DB)



MY_DT				
		Name	Data type	Default value
1		A	Int	0
2		B	Bool	false
3		C	Int	0

DDD					
		Name	Data type	Offset	Start value
1		▼ Static			
2		■ A	Bool	0.0	false
3		■ INT_1	Int	2.0	0
4		■ R	Real	4.0	0.0
5		■ Z	Bool	8.0	false
6		■ GGG	Byte	9.0	16#0
7		■ H	Bool	10.0	false
8		▼ NEW	"MY_DT"	12.0	
9		■ A	Int	12.0	0
10		■ B	Bool	14.0	false
11		■ C	Int	16.0	0

Programovateľný logický automat

Premenné – Elementárne dátové typy a UDT (Príklad z TIA Portal)

TIA Portal – Array v DB (napr. DDD.R[1], DDD.NEW[2,2].A)

DDD					
		Name	Data type	Offset	Start value
1		▼ Static			
2		■ A	Bool	0.0	false
3		■ INT_1	Int	2.0	0
4		■ ▼ R	Array[1..3] of Real	4.0	
5		■ R[1]	Real	4.0	0.0
6		■ R[2]	Real	8.0	0.0
7		■ R[3]	Real	12.0	0.0
8		■ Z	Bool	16.0	false
9		■ GGG	Byte	17.0	16#0
10		■ H	Bool	18.0	false
11		■ ▼ NEW	Array[1..2, 1..2] of *MY_DT*	20.0	
12		■ ▼ NEW[1,1]	*MY_DT*	20.0	
13		■ A	Int	20.0	0
14		■ B	Bool	22.0	false
15		■ C	Int	24.0	0
16		■ ► NEW[1,2]	*MY_DT*	26.0	
17		■ ► NEW[2,1]	*MY_DT*	32.0	
18		■ ► NEW[2,2]	*MY_DT*	38.0	

Programovateľný logický automat

Funkcie a funkčné bloky

- Funkcia (FC) – pre rovnaké vstupy dáva rovnaké výstupy
- Funkčný blok (FB) – na rozdiel od FC obsahuje dátovú pamäť. Výstup závisí od vstupov a vnútorných stavov a preto pre rovnaké vstupy dáva rôzne výstupy. Kód program sa v pamäti nachádza len raz.

Programovateľný logický automat

Funkcie a funkčné bloky

Funkcia je logický blok. Ktorý neobsahuje pamäť. Dáta s ktorými pracuje FC sa ukladajú do lokálneho dátového zásobníka a po ukončení FC sa zmažú. Pri každom volaní funkcie je nutné vždy priradiť formálnym parametrom skutočné parametre. Formálne parametre, ktoré používame priamo vo funkcii, sú iba ukazovateľmi na skutočné parametre logického bloku, z ktorého bola funkcia volaná.

Programovateľný logický automat

Funkcie a funkčné bloky

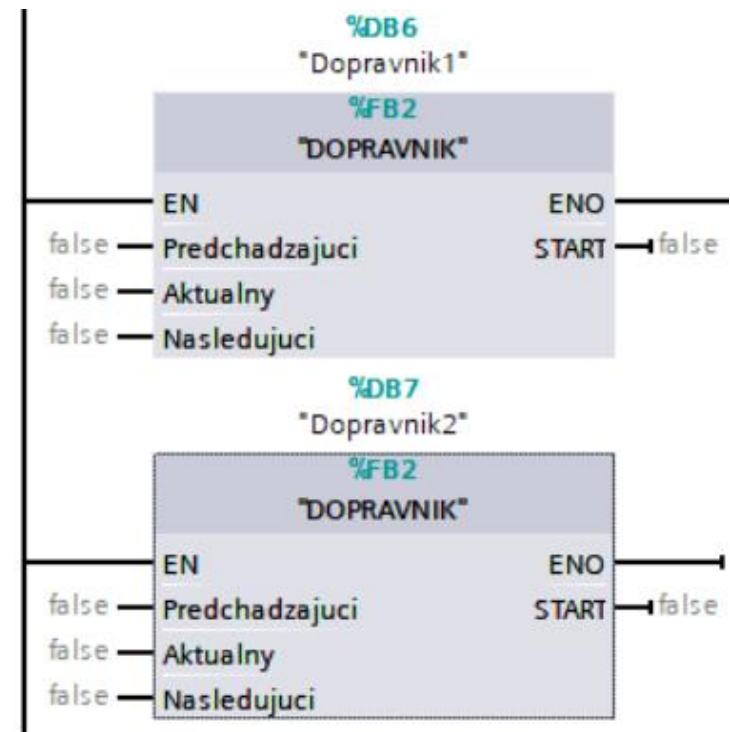
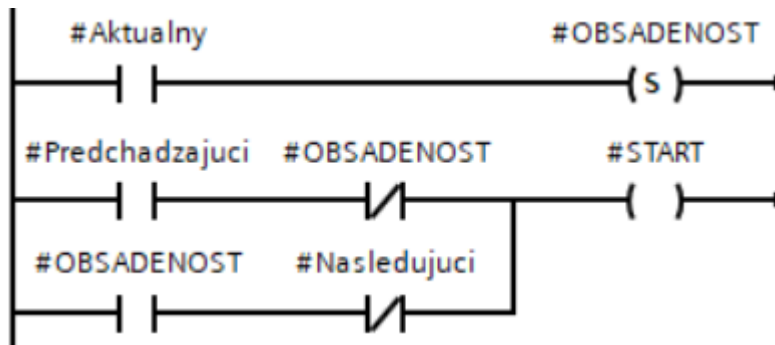
K FB musí byť priradený Inštančný dátový blok. V tomto DB sú uložené všetky statické premenné konkrétneho FB. Dáta uložené v inštančnom DB tam zostanú aj po ukončení vykonávania FB - na rozdiel od dočasných premenných uložených v lokálnom dátovom zásobníku.

Na rozdiel od funkcií FC majú funkčné bloky pamäť. Blok lokálnych dát je priradený k FB. Ak voláme FB, musíme špecifikovať inštančný DB priradený k danému FB. Inštančný dátový blok sa používa k ukladaniu statických premenných. Premenné typu Temp sú uložené v lokálnom dátovom zásobníku. Jeden inštančný DB môže odkazovať len na jeden FB, avšak k jednému FB môže byť priradených viacero DB.

TIA Portal

Prístupy podľa miesta uloženia tagov

1. Tagy (bool, INT, ... , multiinštančné dátové bloky) sú uložené ako statické premenné vo FB. Každá inštancia má svoje stavy v ich dátových blokoch.



TIA Portal

Prístupy podľa miesta uloženia tagov

2. Tagy (stavy objektu) sú uložené v globálnom dátovom bloku (napr. ako štruktúra) a sú pripojené na vstup FB ako inout.

The image shows two screenshots from the TIA Portal software. The top screenshot shows the 'PLC data types' folder in the project tree on the left. A red arrow points from the 'Moje_UDT' data type to a table titled 'Moje_UDT'. The table has four columns: 'Name', 'Data type', and 'Default value'. It lists three variables: 'AUTO' (Bool, false), 'OBSADENOST' (Bool, false), and 'POCET' (Int, 0). The bottom screenshot shows the 'GLOBALNY_DB [DB5]' data block in the project tree on the left. A red arrow points from it to a table titled 'GLOBALNY_DB'. The table has two columns: 'Name' and 'Data type'. It lists three entries: 'Static' (Data type: Static), 'DOPRAVNIKY' (Data type: Array[0..10] of *Moje_UDT*), and '<Add new>'.








	Name	Data type	Default value
1	AUTO	Bool	false
2	OBSADENOST	Bool	false
3	POCET	Int	0

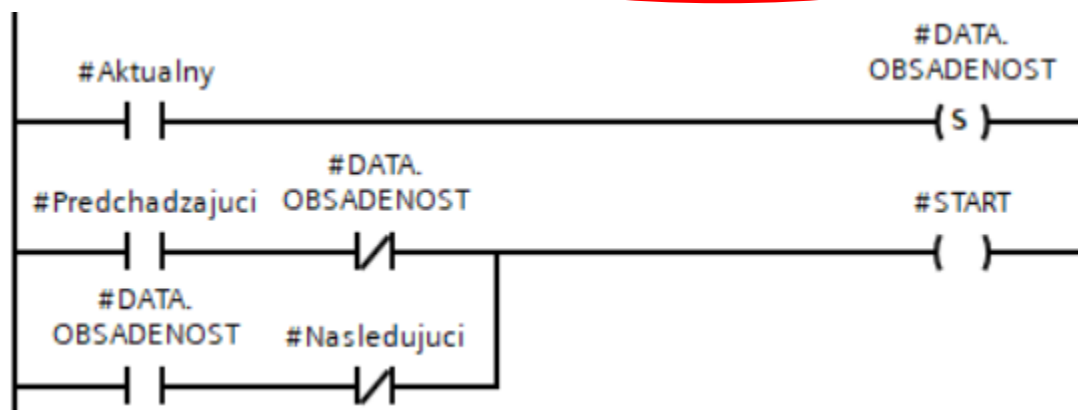
	Name	Data type
1	Static	
2	DOPRAVNIKY	Array[0..10] of *Moje_UDT*
3	<Add new>	

TIA Portal

Prístupy podľa miesta uloženia tagov

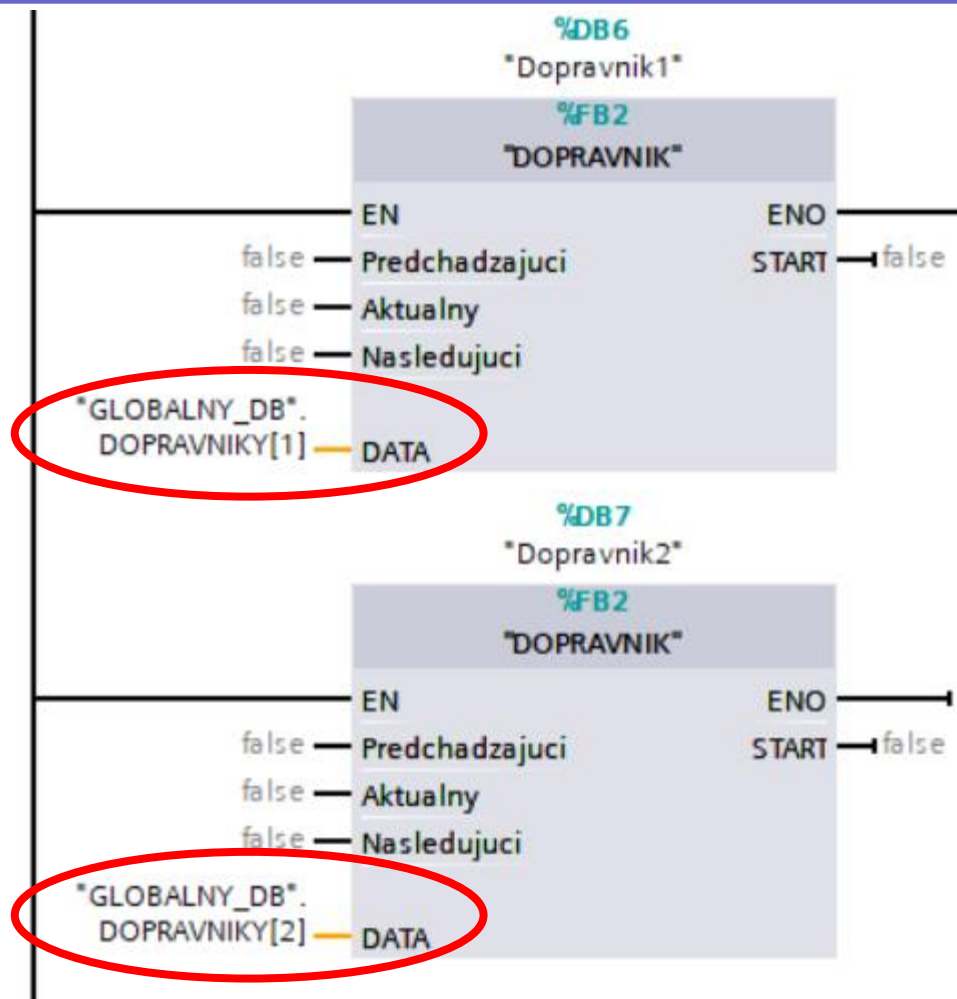
 DOPRAVNIK [FB2]
 NAZOV [FB1]
 Dopravnik1 [DB6]

DOPRAVNIK		
	Name	Data type
1	▼ Input	
2	 Predchadzajuci	Bool
3	 Aktualny	Bool
4	 Nasledujuci	Bool
5	 <Add new>	
6	▼ Output	
7	 START	Bool
8	 InOut	
9	 DATA	*Moje_UDT*



TIA Portal

Prístupy podľa miesta uloženia tagov



Soft computing v radiacích systémoch

Riešenia

- Neuro, fuzzy, ... systém implementovaný v nadradenom systéme (napr. PC alebo iný radiaci systém). Potreba komunikácie PLC-PC (opc, priemyselný ethernet, ...).
- Neuro, fuzzy, ... systém vytvorený v inom prostredí ako PLC (Matlab, softvér na PC, ... Export systému ako objekt (funkčný blok) a import do PLC.
 - Soft computing metóda implementovaná na CPU PLC
 - Soft computing metóda implementovaná pomocou rozširujúceho modulu

Výpočty:

- For cykly podľa ST (IEC štandard)
- Existujúce knižnice pre maticové operácie (napr. Library of general functions (LGF) for SIMATIC STEP 7 (TIA Portal) and SIMATIC S7-1200 / S7-1500)

Soft computing v radiacích systémoch

Existujúce riešenia

Rockwell Automation - RSLogix 5000 Fuzzy Designer
(pre staršie verzie RSLogix 5000)

- Nezávislý nástroj pre tvorbu fuzzy systémov.
- Navrhnutý fuzzy systém vygeneruje tzv. Add-on inštrukciu (v iných RS funkčný blok) v XML formáte.
- Inštrukcia sa importuje do RSLogix 5000.
- Možnosti:
 - CPU ako regulačný člen, supervízor alebo model procesu
 - Diagnostika – klasifikácia stavov procesu
 - Rozhodovací systém pre operátorov
 - Predikcia stavov / veličín

Soft computing v radiacích systémoch

Existujúce riešenia

Siemens - Neural Processing Unit (NPU)

- Rozširujúci modul pre PLC (nezáv. od CPU)
- AI-capable MyriadTMX Vision processing unit chip from Intel MovidiusTM
- Pripojenie kamery, mikrofónu, ...
- Open AI framework ako Tensorflow, Caffe
- Postup:
 - Rozšírenie HW konf. o nový modul (inštalácia HSP)
 - Vytvorenie programu pomocou frameworku
 - Inštalácia (prenos) projektu na pamäťovú kartu NPU modulu
 - Komunikácia CPU <-> NPU pomocou FB

Soft computing v radiacích systémoch

Existujúce riešenia

Siemens – Edge (vid' TP 😊) – inference server

- Priemyselné PC špeciálne pre AI (zatiaľ na báze CPU, ale chystajú sa aj s grafikou)
- Založené na Docker prístupe

Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Zadanie bakalárskej práce:

Cieľom bakalárskej práce je naštudovať problematiku fuzzy systémov pre oblasť modelovania a riadenia procesov a navrhnúť všeobecný fuzzy toolbox pre Siemens PLC.

Úlohy:

1. Naštudujte teóriu z oblasti fuzzy systémov.
2. Naštudujte programovací nástroj TIA Portal.
3. Navrhnite a implementujte fuzzy systém pre vybraný PLC systém.
4. Navrhnite a implementujte fuzzy systém ako regulačný člen hydraulickej sústavy.
5. Overte funkčnosť navrhnutého fuzzy systému a výsledky vhodne vyhodnoťte.
6. Vytvorte jednoduchú dokumentáciu.

Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

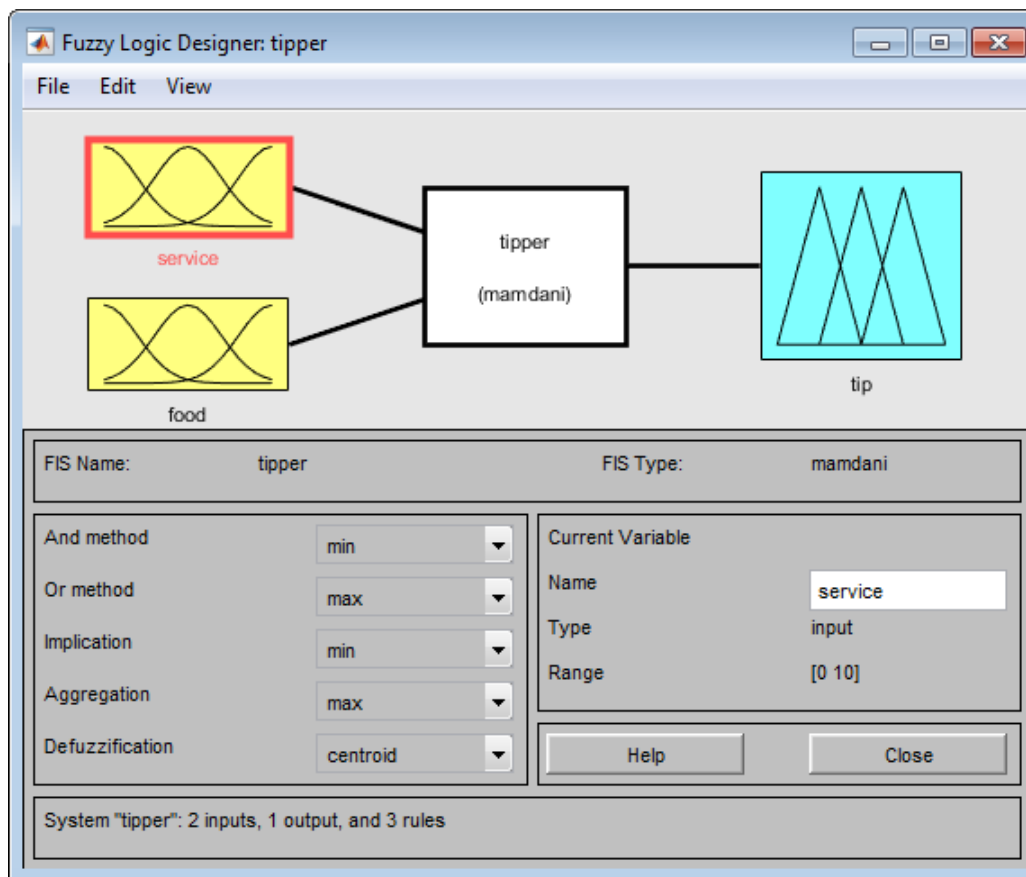
Podciele:

- Kompatibilita s Matlabom
- Možnosť exportu z Matlabu (nasadenie v RS)
- Všeobecná štruktúra v PLC (ako model, riadiaci člen, ...)

Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

1. Vytvorenie fuzzy systému pomocou Fuzzy Logic Toolboxu



Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

```
nase_fuzzy =
```

```
mamfis with properties:
```

```
    Name: "nase_fuzzy"
```

```
    AndMethod: "min"
```

```
    OrMethod: "max"
```

```
    ImplicationMethod: "min"
```

```
    AggregationMethod: "max"
```

```
    DefuzzificationMethod: "centroid"
```

```
    Inputs: [1×2 fisvar]
```

```
    Outputs: [1×1 fisvar]
```

```
    Rules: [1×2 fisrule]
```

```
    DisableStructuralChecks: 0
```

Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

```
>> nase_fuzzy.AndMethod
```

```
ans = "min"
```

```
>> nase_fuzzy.Inputs
```

```
ans = 1×2 fisvar array with properties:
```

Name

Range

MembershipFunctions

Details:

Name	Range	MembershipFunctions
"input1"	0 1	{1×3 fismf}
"input2"	0 1	{1×3 fismf}

Soft computing v radiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

```
>> nase_fuzzy.Inputs(1).Name
```

```
ans =    "input1"
```

```
>> nase_fuzzy.Inputs(1).Range
```

```
ans =    0    1
```

```
>>nase_fuzzy.Inputs(1).Range(1)
```

```
ans =    0
```

```
>> nase_fuzzy.Inputs(1).Range(2)
```

```
ans =    1
```














```
atd'.
```

Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

2. Vytvorenie XML súboru pre TIA Portal (pre dátový blok)

Majme nasledovný dátový blok v TIA Portal:

DATA					
		Name	Data type	Start value	Retain
1		▼ Static			<input type="checkbox"/>
2		■ INT_1	Int	5	<input type="checkbox"/>
3		■ REAL_1	Real	1.2	<input type="checkbox"/>
4		■ ▼ Vektor	Array[0..1] of Int		<input type="checkbox"/>
5		■ Vektor[0]	Int	7	<input type="checkbox"/>
6		■ Vektor[1]	Int	8	<input type="checkbox"/>
7		■ ▼ Vektor2	Array[0..1] of Struct		<input checked="" type="checkbox"/>
8		■ ▼ Vektor2[0]	Struct		<input type="checkbox"/>
9		■ CISLO	Int	10	<input type="checkbox"/>
10		■ BIT	Bool	TRUE	<input type="checkbox"/>
11		■ <Add new>			<input type="checkbox"/>
12		■ ▼ Vektor2[1]	Struct		<input type="checkbox"/>
13		■ CISLO	Int	11	<input type="checkbox"/>
14		■ BIT	Bool	FALSE	<input type="checkbox"/>

Soft computing v radiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Exportom DB (pravý klik na DB) získame XML štruktúru (bez manuálu)



```
DATA_BLOCK "DATA"  
{ S7_Optimized_Access := 'TRUE' }  
VERSION : 0.1  
NON_RETAIN  
VAR  
  INT_1 : Int;  
  REAL_1 : Real;  
  Vektor : Array[0..1] of Int;  
  Vektor2 : Array[0..1] of Struct  
    CISLO : Int;  
    BIT : Bool;  
  END_STRUCT;  
END_VAR
```

Soft computing v radiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

```
BEGIN
```

```
  INT_1 := 5;
```

```
  REAL_1 := 1.2;
```

```
  Vektor[0] := 7;
```

```
  Vektor[1] := 8;
```

```
  Vektor2[0].CISLO := 10;
```

```
  Vektor2[0].BIT := TRUE;
```

```
  Vektor2[1].CISLO := 11;
```

```
  Vektor2[1].BIT := FALSE;
```

```
END_DATA_BLOCK
```

Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Časť zdrojového kódu súboru matlab_to_tia.m:

```
% ODLADENIE NEVHODNEHO NAVRHU FUZZY SYSTEMU - mamdani, centroid/mom
```

```
% min 1 input, min 1 output, min 1 rule, trimf/trapmf/gaussmf
```

```
fuzzyToTIA(logika,"fuzzyDB.db")
```

```
%logika - nazov premennej s fuzzy logikou, trieda 'mamfis'
```

```
%menoSuboru - nazov vystupneho suboru, musi byt koncovka .db"
```

```
function fuzzyToTIA(logika, menoSuboru)
```

```
    nazov = "fuzzyDB";
```

```
    velkost = length(logika);
```

```
    for z = 1:velkost
```

```
        paramCount = 4;
```

```
        if (~strcmp(logika(z).type,'mamdani'))
```

```
            fprintf('Fuzzy system musi byt typu mamdami!\r\n');
```

```
            return;
```

```
        elseif (~strcmp(logika(z).defuzzMethod,'centroid'))
```

```
            if (~strcmp(logika(z).defuzzMethod,'mom'))
```

```
                fprintf('Zvol inu metodu defuzzifikacie - centroid/mom!\r\n');
```

Soft computing v riadiacich systémoch

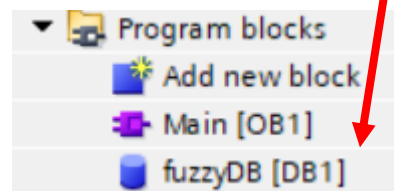
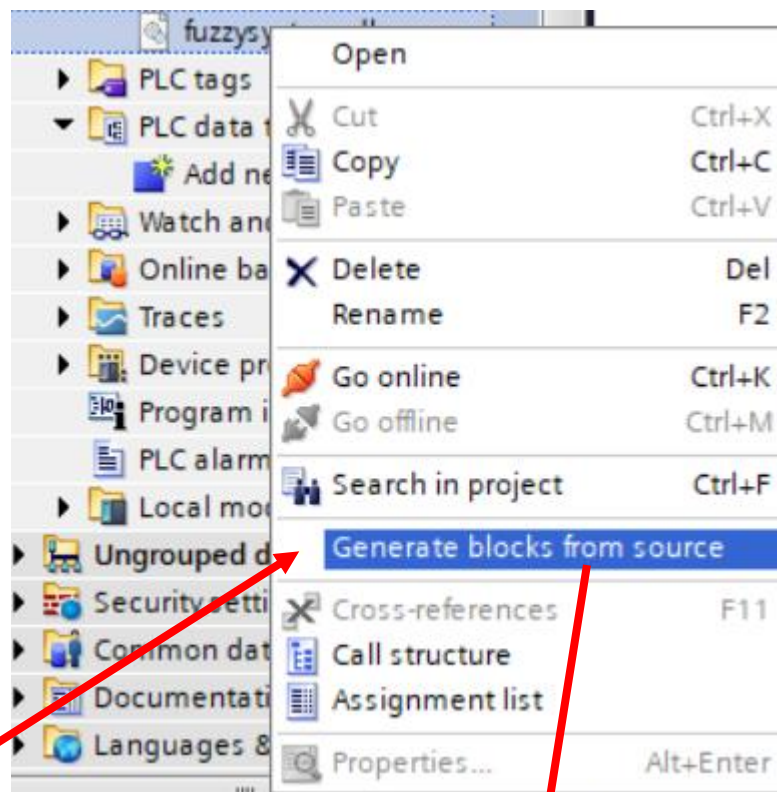
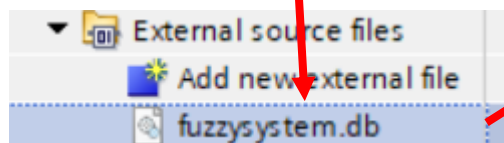
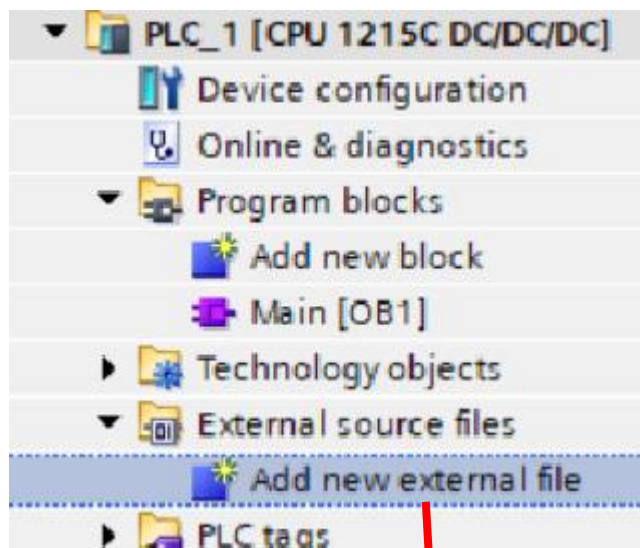
Fuzzy systém pre Siemens PLC (Michal Jakabovič)

```
fid = fopen(menoSuboru,'w+'); % zadanie nazvu pre subor .db
fprintf(fid,'DATA_BLOCK "%s"\r\n', nazov); % zadanie nazvu datoveho bloku
fprintf(fid,{ S7_Optimized_Access := "TRUE" }\r\n');
fprintf(fid,'VERSION : 0.1\r\n');
fprintf(fid,'NON_RETAIN\r\n');
fprintf(fid,'VAR\r\n');
fprintf(fid,'logika : Array[1..%d] of Struct\r\n',velkost);
fprintf(fid,'NumInputs : int;\r\n');
fprintf(fid,'NumOutputs : int;\r\n');
fprintf(fid,'NumRules : int;\r\n');
fprintf(fid,'AndMethod : String;\r\n');
fprintf(fid,'OrMethod : String;\r\n');
fprintf(fid,'ImpMethod : String;\r\n');
fprintf(fid,'AggMethod : String;\r\n');
fprintf(fid,'DefuzzMethod : String;\r\n');
fprintf(fid,'Input : Array[1..%d] of Struct\r\n',vstupy);
fprintf(fid,'Range : Array[1..2] of Real;\r\n');
```


Soft computing v riadiacich systémoch
























Fuzzy systém pre Siemens PLC (Michal Jakabovič)

3. Import dátového bloku



Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

fuzzyDB				
		Name	Data type	Start value
1		▼ Static		
2		■ ▼ logika	Array[1..1] of Struct	
3		■ ▼ logika[1]	Struct	
4		■ NumInputs	Int	2
5		■ NumOutputs	Int	1
6		■ NumRules	Int	25
7		■ AndMethod	String	'min'
8		■ OrMethod	String	'max'
9		■ ImpMethod	String	'min'
10		■ AggMethod	String	'max'
11		■ DefuzzMethod	String	'centroid'
12		■ ▼ Input	Array[1..2] of Struct	
13		■ ▼ Input[1]	Struct	
14		■ ▼ Range	Array[1..2] of Real	
15		■ Range[1]	Real	-2.0
16		■ Range[2]	Real	2.0
17		■ NumMFs	Int	5
18		■ mf_in	Real	0.0
19		■ ▶ MF	Array[1..5] of Struct	
20		■ ▶ Input[2]	Struct	
21		■ ▶ Output	Array[1..1] of Struct	
22		■ ▶ Rule	Array[1..25] of Struct	

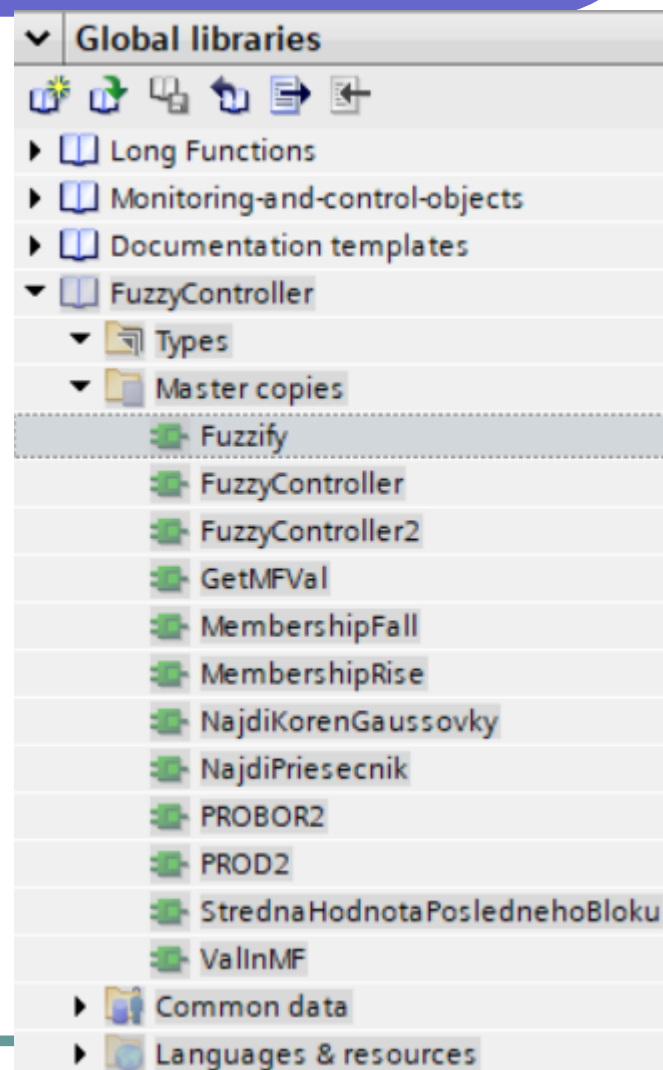
Soft computing v radiacích systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

4. Implementácia fuzzy systému

- *.ap15_1 – súbor, ktorým sa spúšťa (otvára) projekt v15.1
- *.zap15_1 – archív projektu
- *.al15_1 – knižnica (library file resp. projekt)











Opíšeme vybrané funkcie...



Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Fuzzify – funkcia slúžiaca na výpočet hodnôt počas fuzzyfikácie. Má dva parametre a to ostrú hodnotu a typ funkcie.

Fuzzify			
		Name	Data type
1		▼ Input	
2		■ val	Real
3		■ ▼ MF	Struct
4		■ Typ	String
5		■ ▼ Params	Array[1..4] of Real
6		■ Params[1]	Real
7		■ Params[2]	Real
8		■ Params[3]	Real
9		■ Params[4]	Real
10		■ out	Real

Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)





```
IF #MF.Typ = 'trimf' THEN
  REGION Trimf
    // the membership function has a triangular shape
    IF (#val >= #MF.Params[1]) AND (#val <= #MF.Params[2]) THEN
      // rising edge of the triangle
      #Fuzzify := "MembershipRise"(input := #val, min := #MF.Params[1], max := #MF.Params[2]);
    ELSIF (#val >= #MF.Params[2]) AND (#val <= #MF.Params[3]) THEN
      // falling edge of the triangle
      #Fuzzify := "MembershipFall"(input := #val, min := #MF.Params[2], max := #MF.Params[3]);
    ELSE
      #Fuzzify := 0;
    END_IF;
  END_REGION
```

...

Soft computing v radiacích systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

MembershipRise – vypočíta hodnotu funkcie príslušnosti priamky

MembershipRise			
		Name	Data type
1		▼ Input	
2		input	Real
3		min	Real
4		max	Real













```
#rangeSize := #max - #min;
```

```
#MembershipRise := ( #input - #min ) / #rangeSize;
```

Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

GetMFVal – funkcia, ktorá pomocou správnej implikačnej metódy vyhodnotí výsledok pravidla

GetMFVal					
		Name	Data type	Default value	Comment
1		▼ Input			
2		■ val	Real		Xova hodnota ktorej pravdepodobnost chceme najst
3		■ ▼ MF	Struct		Pravidlo
4		■ Typ	String		
5		■ ▼ Params	Array[1..4] of Real		
6		■ Params[1]	Real		
7		■ Params[2]	Real		
8		■ Params[3]	Real		
9		■ Params[4]	Real		
10		■ out	Real		
11		■ metoda	String		metoda implikacie
12		■ RuleVal	Real		hodnota pravidla

Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

```
#GetMFVal := 0;

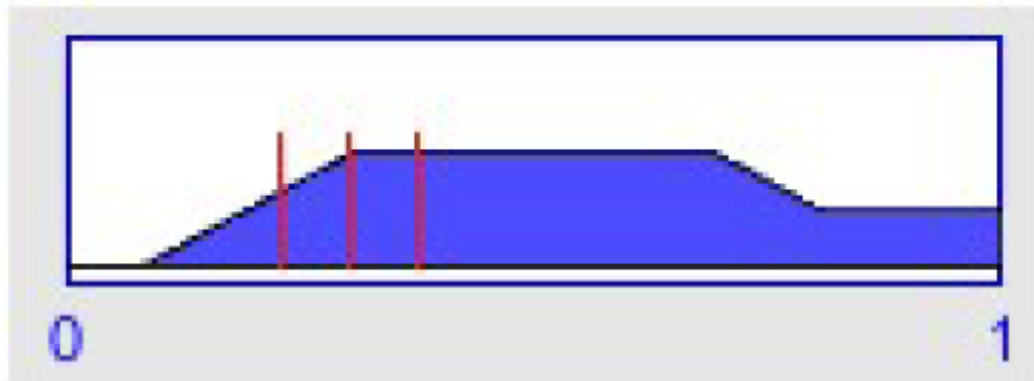
IF #metoda = 'min' THEN
    #GetMFVal:= MIN(IN1:=#RuleVal, IN2:"Fuzzify"(val:= #val, MF:= #MF));
ELSIF #metoda = 'prod' THEN
    #GetMFVal:= "PROD2"(a:= #RuleVal, b:"Fuzzify"(val:= #val, MF:= #MF) );
ELSE
    ; //nepodporovana metoda
END_IF;
RETURN;
```


Soft computing v radiacích systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Krokovací defuzzifikačný algoritmus

- S pevným krokom sa zistia výšky a aplikuje sa metóda defuzzyfikácie. Obdĺžniková náhrada.

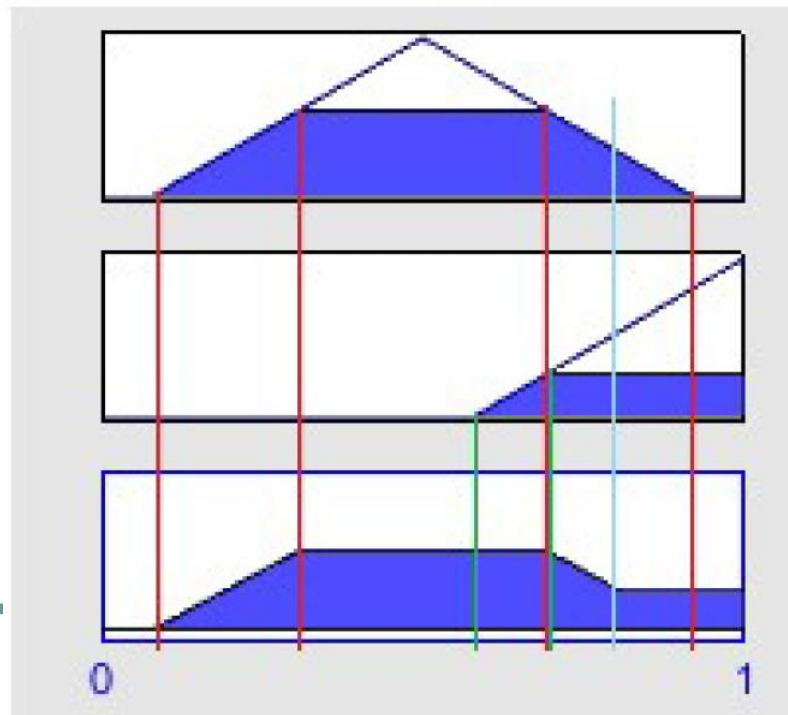


Soft computing v radiacích systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Variabilný defuzzifikačný algoritmus

- Prvým krokom je nájdenie okrajových bodov každej funkcie príslušnosti výstupnej premennej podľa metódy implikácie pravidiel. Následným krokom je vytvorenie zoznamu týchto okrajových bodov a ich zoradením. Súčasťou zoznamu okrajových bodov sú aj pravidlá, ku ktorým okrajové body patria ako je znázornené na obrázku



Soft computing v radiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Príklad: Hydraulická sústava

$$S \frac{\delta h}{\delta t} = Q_i - \mu S_0 \sqrt{2gh}$$

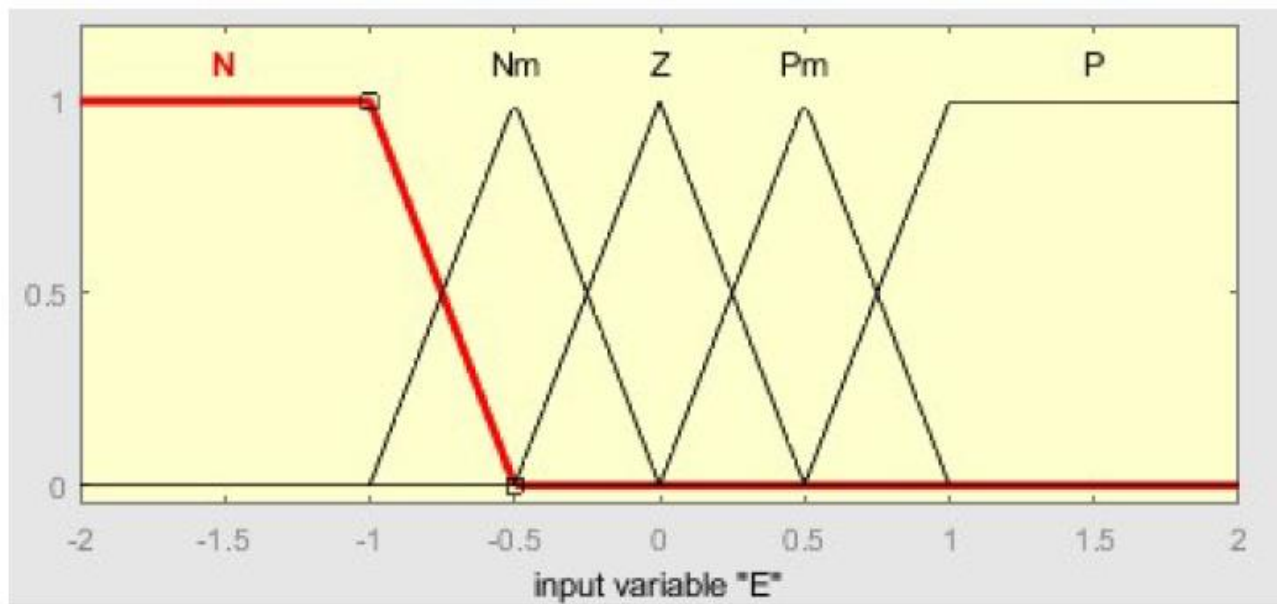
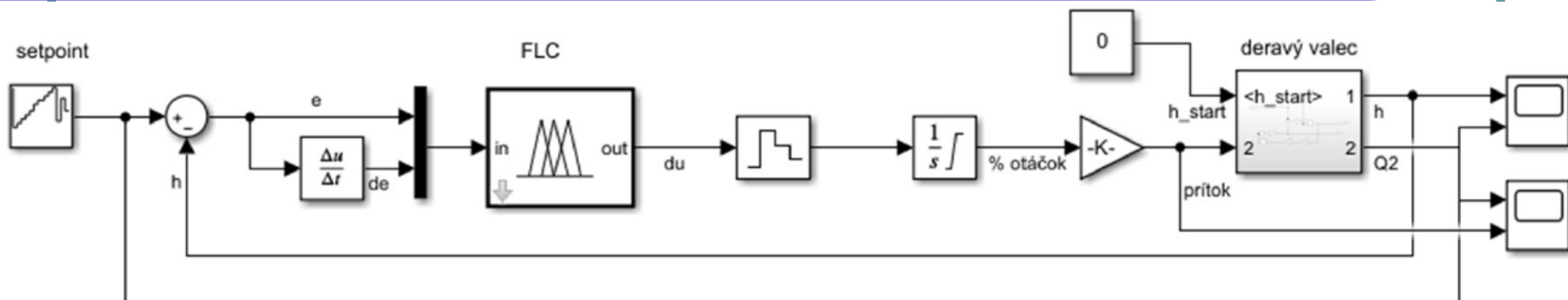
$$S = 2\text{m}^2, S_0 = 0.02\text{m}^2, \mu = 0.63, g = 0.81\text{m/s}^2$$

$$h_{\max} = 2\text{m}, Q_{\max} = 0.4 \text{ m}^3$$

$$\Rightarrow h \in \langle 0\text{m}, 2\text{m} \rangle, dh \in \langle -2\text{m}, 2\text{m} \rangle, e \in \langle -2\text{m}, 2\text{m} \rangle$$

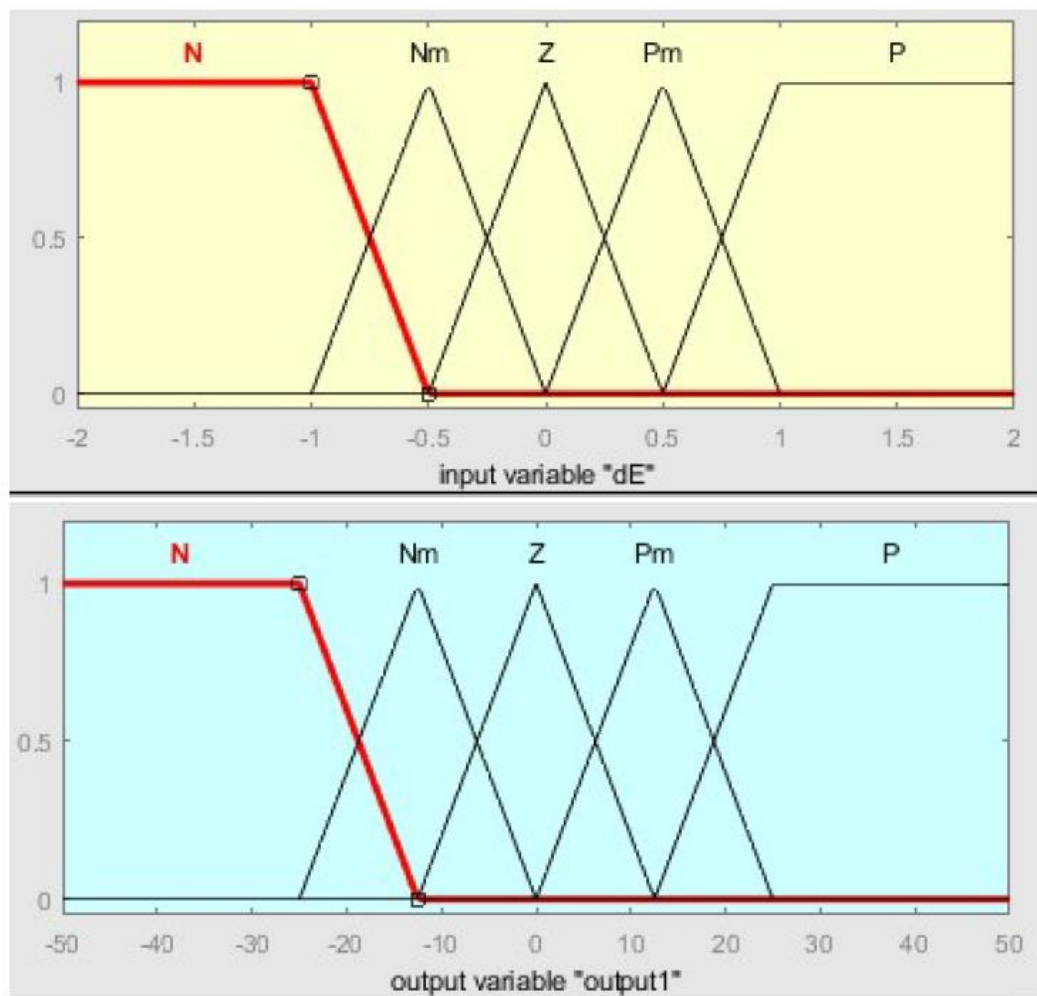
Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)



Soft computing v radiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)



Soft computing v radiacich systémoch

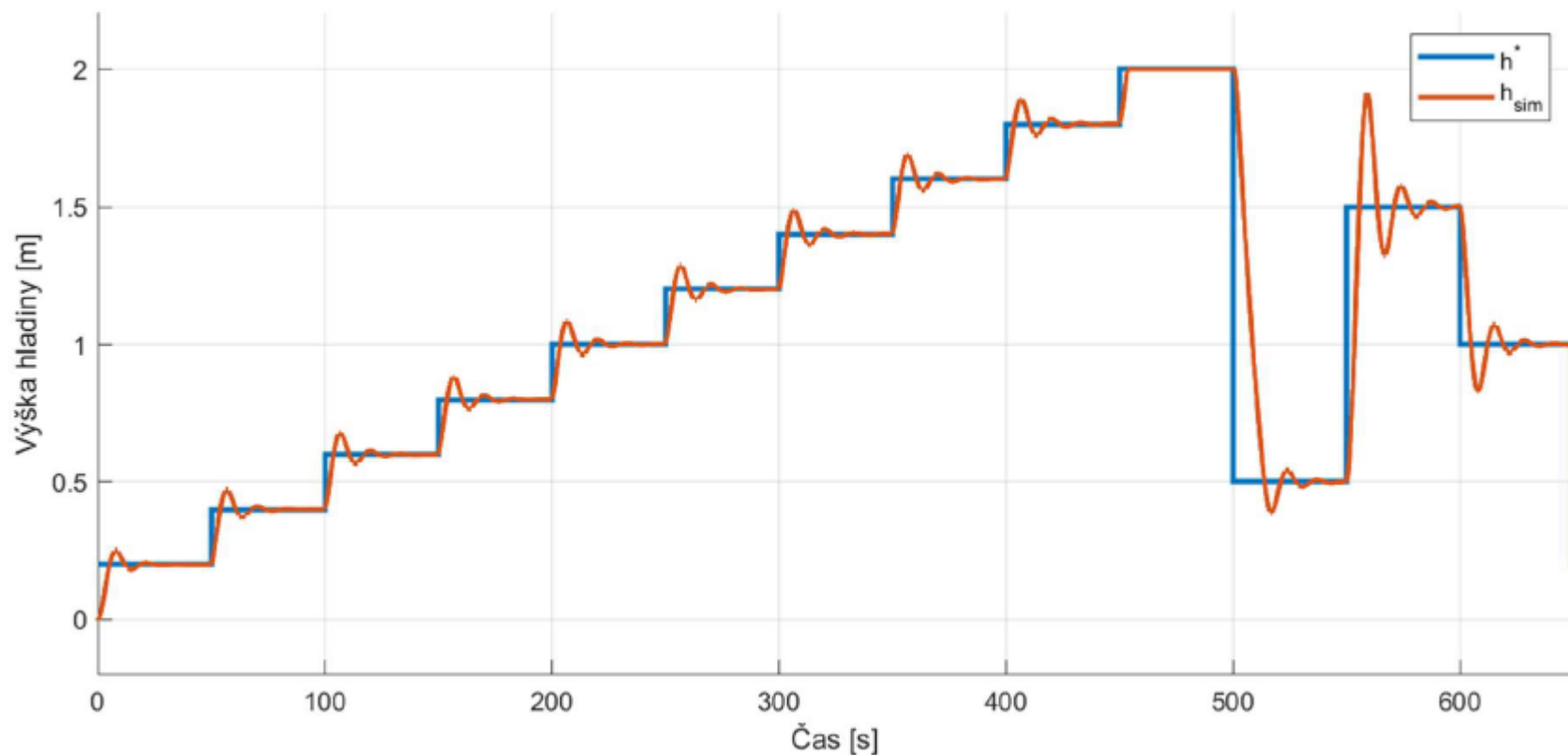
Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Štandardná báza pravidiel ako náhrada PI, PID, .. regulátora

ΔE E	N	Nm	Z	Pm	P
N	N	N	Nm	Nm	Z
Nm	N	Nm	Nm	Z	Pm
Z	Nm	Nm	Z	Pm	Pm
Pm	Nm	Z	Pm	Pm	P
P	Z	Pm	Pm	P	P

Soft computing v radiacích systémech

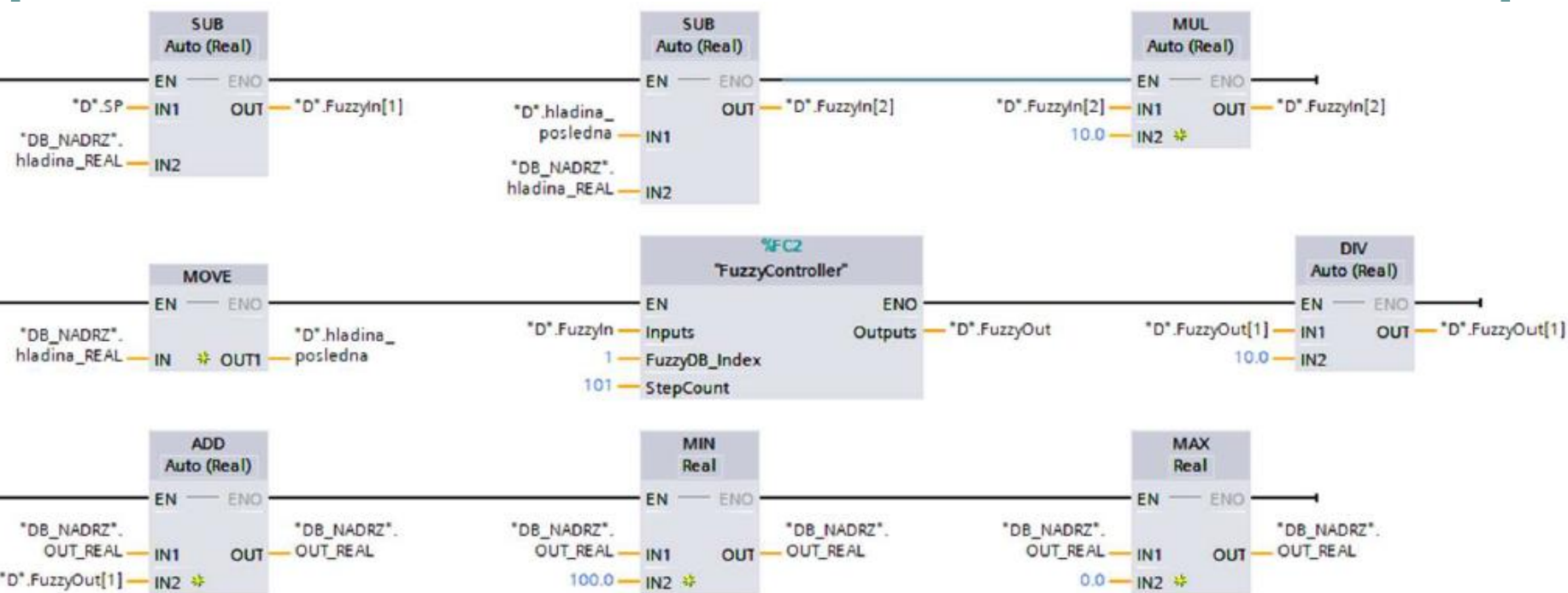
Fuzzy systém pre Siemens PLC (Michal Jakabovič)



Soft computing v riadiacich systémoch

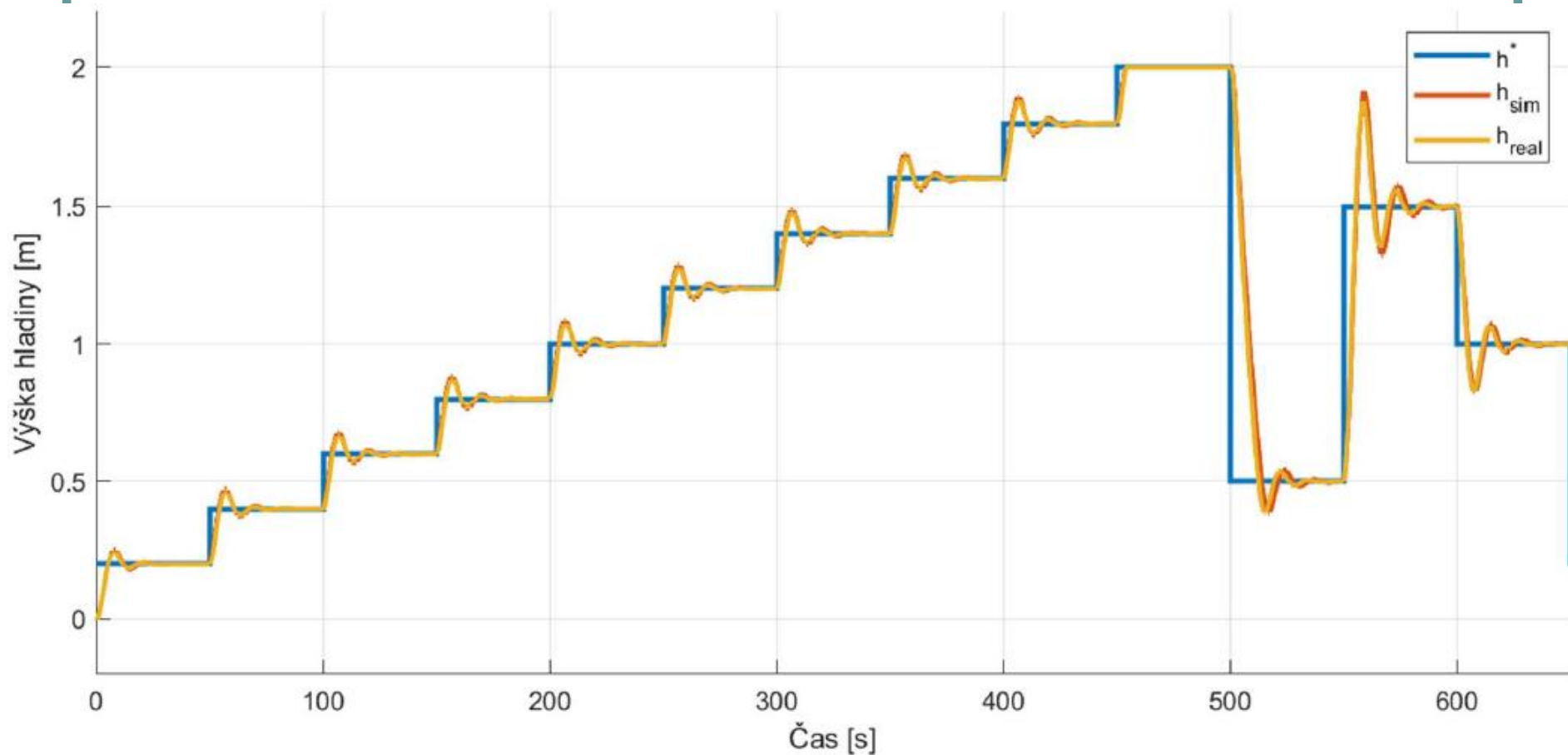
Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Implementácia v PLC (predspracovanie a FuzzyController)



Soft computing v radiacích systémech

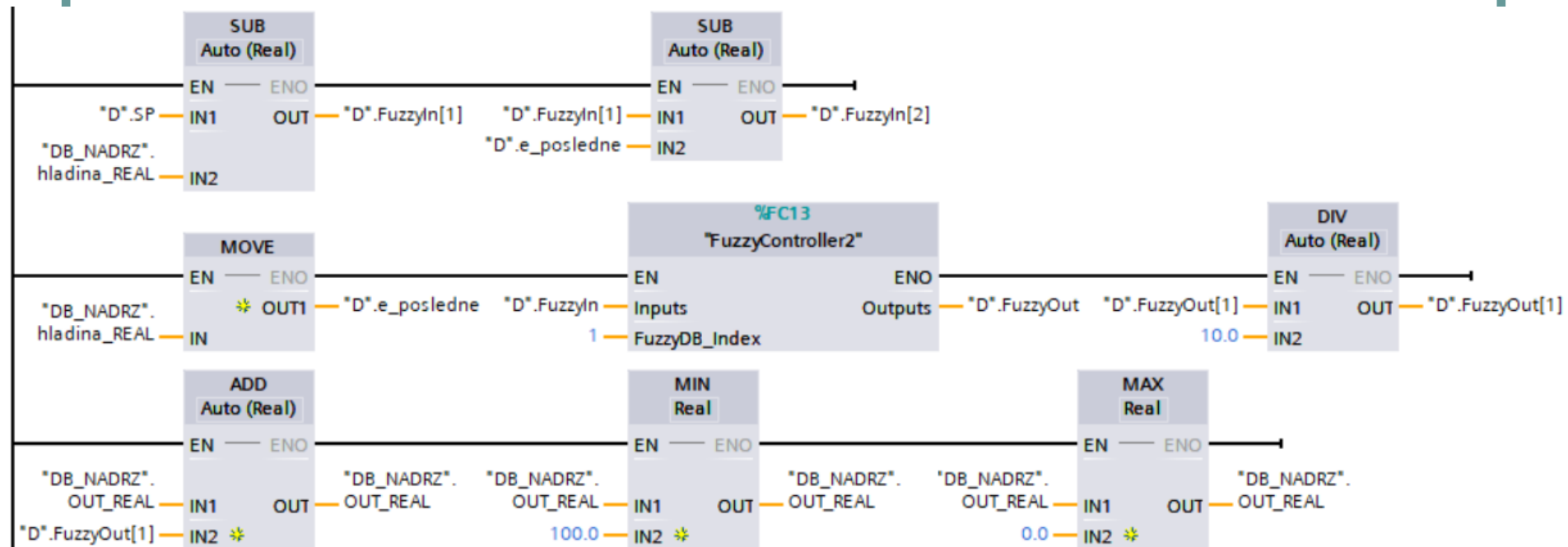
Fuzzy systém pre Siemens PLC (Michal Jakabovič)



Soft computing v riadiacich systémoch

Fuzzy systém pre Siemens PLC (Michal Jakabovič)

Opravený výpočet 2. vstupu pre fuzzy systém.



Soft computing v radiacích systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Zadanie bakalárskej práce:

Cieľom bakalárskej práce je návrh, implementácia a overenie všeobecnej štruktúry umelej neurónovej siete (UNS) perceptrón pre Siemens PLC.

Úlohy:

1. Naštudujte základy teórie z oblasti UNS.
2. Naštudujte softvér TIA Portal na programovanie a konfiguráciu Siemens PLC.
3. Naštudujte základy neurónového toolboxu v Matlabe.
4. Navrhните všeobecnú štruktúru UNS perceptrón (počet vstupných neurónov, počet výstupných neurónov, počet skrytých vrstiev, počet neurónov v skrytých vrstvách, aktivačné funkcie atď.).
5. Implementujte a overte navrhnutú štruktúru pre vybrané PLC. Porovnanie vykonajte s Matlabom. Implementácia nemá obsahovať metódu učenia UNS.
6. Výsledky vhodne vyhodnoťte a vytvorte vhodnú dokumentáciu.

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Podciele:

- Kompatibilita s Matlabom
- Možnosť exportu z Matlabu (nasadenie v RS)
- Všeobecná štruktúra v PLC (ako model, riadiaci člen, ...)

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Model: Hydraulický systém s rovnakými parametrami.

Matlab: Neural network toolbox (funkcia newff – naposledy R2010a NNET 6.0.4; feedforwardnet – v novších verziách).

Export UNS: Súbor generujúci dátový blok vo forme XML.

Niektoré časti premennej Net obsahujúca všetko o UNS:

Net.layers.transferfcn – aktivačné funkcie

Net.numberlayers – počet vrstiev

Net.iw – váhy medzi vstupnou a prvou skrytou vrstvou

Net.lw – váhy medzi skrytou a výstupnou vrstvou

Net.b - biasy

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Demo príklad (Matlab):

```
[x,t] = simplefit_dataset;
```

```
net = feedforwardnet(10);
```

```
net.numInputs
```

```
net.numLayers
```

```
>> net.IW
```

```
ans = 2×1 cell array {10×0 double} { 0×0 double}
```

```
net.IW{1}
```

```
net.IW{2}
```

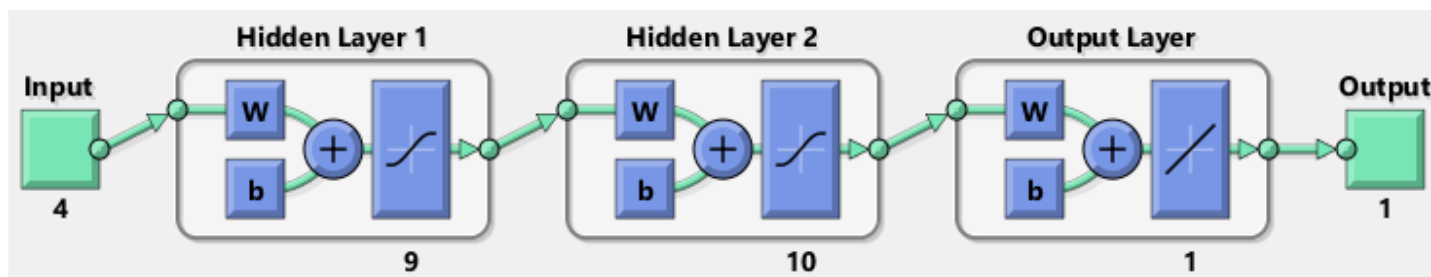
```
...
```

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

UNS pre hydraulickú sústavu:

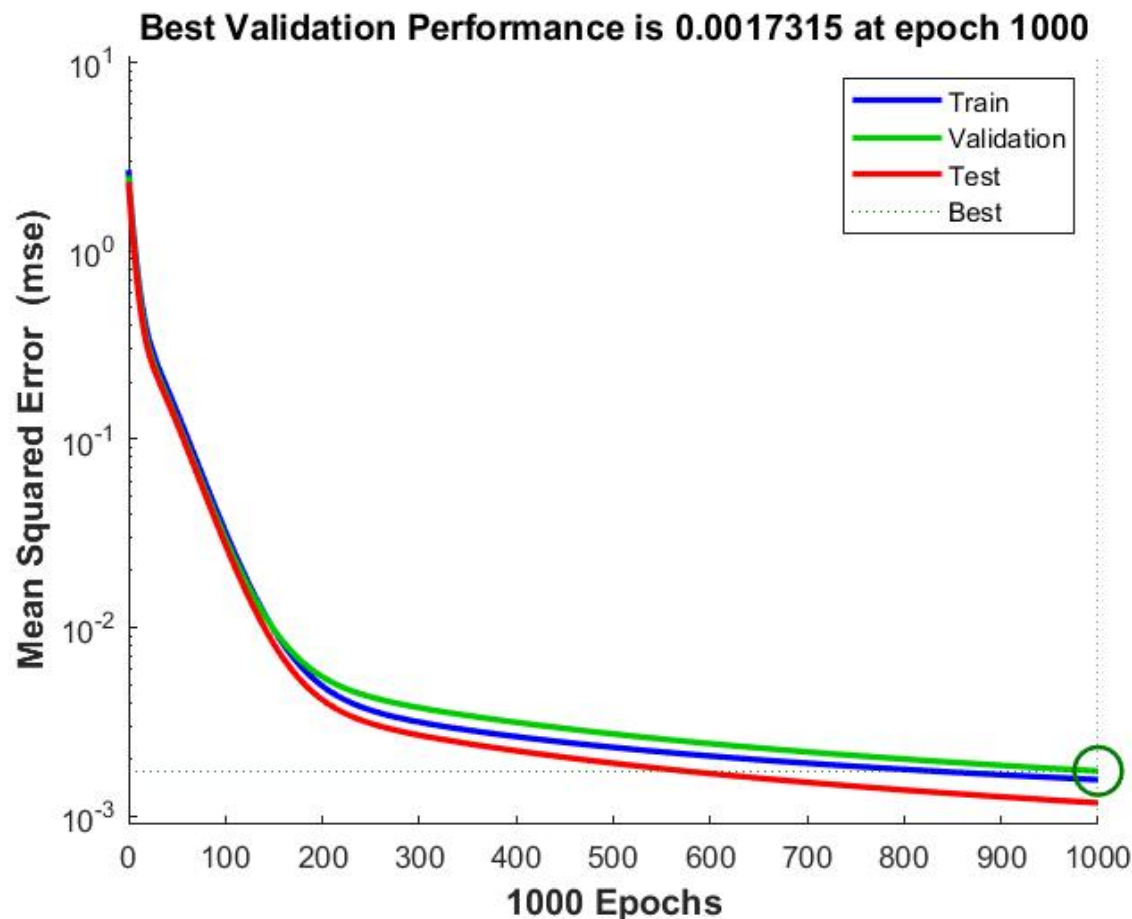
- 4 vstupy reprezentujúce $x[k-2]$, $x[k-1]$, $y[k-2]$ a $y[k-1]$
- 1 výstup reprezentujúci $y[k]$
- 9 a 10 neurónov v 2 skrytých vrstvách (tansig)
- metóda trénovania: traingd
- počet epoch: 1000
- rozdelenie dát na trénovacie, validačné a testovacie



Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Trénovanie...



Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Časť zdrojového kódu súboru StrukturaUNS.m:

```
% VYTVARANIE Programu PRE TIA PORTAL V13
```

```
file = fopen('NeuralNetwork.db','w+'); % zadanie nazvu pre subor .db
```

```
fprintf(file,'DATA_BLOCK "NeuralNetwork"\r\n'); % zadanie nazvu datoveho bloku
```

```
fprintf(file,'{ S7_Optimized_Access := "TRUE" }\r\n');
```

```
fprintf(file,'VERSION : 0.1\r\n');
```

```
fprintf(file,'NON_RETAIN\r\n');
```

```
fprintf(file,'VAR\r\n');
```

```
fprintf(file,'NumInputs : int;\r\n');
```

```
fprintf(file,'NumOutputs : int;\r\n');
```

```
fprintf(file,'NumHiddenLayers : int;\r\n');
```

```
fprintf(file,'NumLayers : int;\r\n');
```

```
fprintf(file,'Iteration : int;\r\n');
```

```
fprintf(file,'IterationTest : int;\r\n');
```

```
fprintf(file,'Rare : real;\r\n');
```

```
fprintf(file,'Epoch : int;\r\n');
```

```
...
```

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

```
...  
fprintf(file,'NumInputs := %d;\r\n',net.input.size);  
fprintf(file,'NumOutputs := %d;\r\n',net.output.size);  
fprintf(file,'NumHiddenLayers := %d;\r\n',net.numLayers-1);  
fprintf(file,'NumLayers := %d;\r\n',net.numLayers+1);  
  
fprintf(file,'Iteration := %d;\r\n',numbertrening);  
fprintf(file,'IterationTest := %d;\r\n',numbertest);  
fprintf(file,'Rare := %f;\r\n',net.trainparam.lr);  
fprintf(file,'Epoch := %d;\r\n',net.trainparam.epochs);  
...
```

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Časť dátového bloku v PLC:

NeuralNetwork				
		Name	Data type	Start value
1		▼ Static		
2		■ NumInputs	Int	4
3		■ NumOutputs	Int	1
4		■ NumHiddenLayers	Int	2
5		■ NumLayers	Int	4
6		■ Iteration	Int	2440
7		■ IterationTest	Int	487
8		■ Rare	Real	0.01
9		■ Epoch	Int	1000
10		▸ Minmaxinput	Array[1..1] of Struct	
11		▸ Minmaxoutput	Array[1..1] of Struct	
12		▸ Input	Array[1..2440] of Struct	
13		▸ InputTest	Array[1..487] of Struct	
14		▸ Target	Array[1..2440] of Struct	
15		▸ TargetTest	Array[1..487] of Struct	
16		▸ Bias	Array[2..4] of Struct	
17		▸ Layer	Array[1..4] of Struct	

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

10		▼ Minmaxinput	Array[1..1] of Struct	
11		▼ Minmaxinput[1]	Struct	
12		▼ Xmin	Array[1..4] of Struct	
13		▼ Xmin[1]	Struct	
14		Value	LReal	1.0
15		► Xmin[2]	Struct	
16		► Xmin[3]	Struct	
17		► Xmin[4]	Struct	
18		▼ Xmax	Array[1..4] of Struct	
19		▼ Xmax[1]	Struct	
20		Value	LReal	19.5
21		► Xmax[2]	Struct	
22		► Xmax[3]	Struct	
23		► Xmax[4]	Struct	
24		Ymin	LReal	-1.0
25		Ymax	LReal	1.0
26		► Minmaxoutput	Array[1..1] of Struct	

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Časť zdrojového kódu UNS v PLC:

```
IF #epoch <= "NeuralNetwork".Epoch THEN
```

```
  #iteration := #iteration + 1;
```

```
  IF #iteration = "NeuralNetwork".Iteration + 1 THEN
```

```
    #iteration := 1;
```

```
    #epoch := #epoch + 1;
```

```
    #totalerror := 0;
```

```
  END_IF;
```

```
FOR #input := 1 TO "NeuralNetwork".NumInputs DO
```

```
  "NeuralNetwork".Layer[1].Neural[#input].Value := "NeuralNetwork".Input[#iteration].Neural[#input].Value;
```

```
END_FOR;
```

```
FOR #minmax := 1 TO "NeuralNetwork".NumInputs DO
```

```
  "NeuralNetwork".Layer[1].Neural[#minmax].Value := ((("NeuralNetwork".Input[#iteration].Neural[#minmax].Value -  
"NeuralNetwork".Minmaxinput[1].Xmin[#minmax].Value) * ("NeuralNetwork".Minmaxinput[1].Ymax  
"NeuralNetwork".Minmaxinput[1].Ymin) / ("NeuralNetwork".Minmaxinput[1].Xmax[#minmax].Value -  
"NeuralNetwork".Minmaxinput[1].Xmin[#minmax].Value))) + "NeuralNetwork".Minmaxinput[1].Ymin;
```

```
END_FOR;...
```

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

```
//For cyklus pre výpočet hodnôt neurónov v neurónovej sieti
FOR #layers := 2 TO "NeuralNetwork".NumLayers DO

    FOR #neural1 := 1 TO "NeuralNetwork".Layer[#layers].Dimensions DO

        FOR #neural := 1 TO "NeuralNetwork".Layer[#layers - 1].Dimensions DO
            #tmp      :=      "NeuralNetwork".Layer[#layers      -      1].Neural[#neural].Value      *
            "NeuralNetwork".Layer[#layers].Neural[#neural1].Weight[#neural];
            #sum := #sum + #tmp;
        END_FOR;
        #intrinsicactivity := #sum + "NeuralNetwork".Bias[#layers].Neural[#neural1].Weight;
        "NeuralNetwork".Layer[#layers].Neural[#neural1].Sensitivities := #intrinsicactivity;

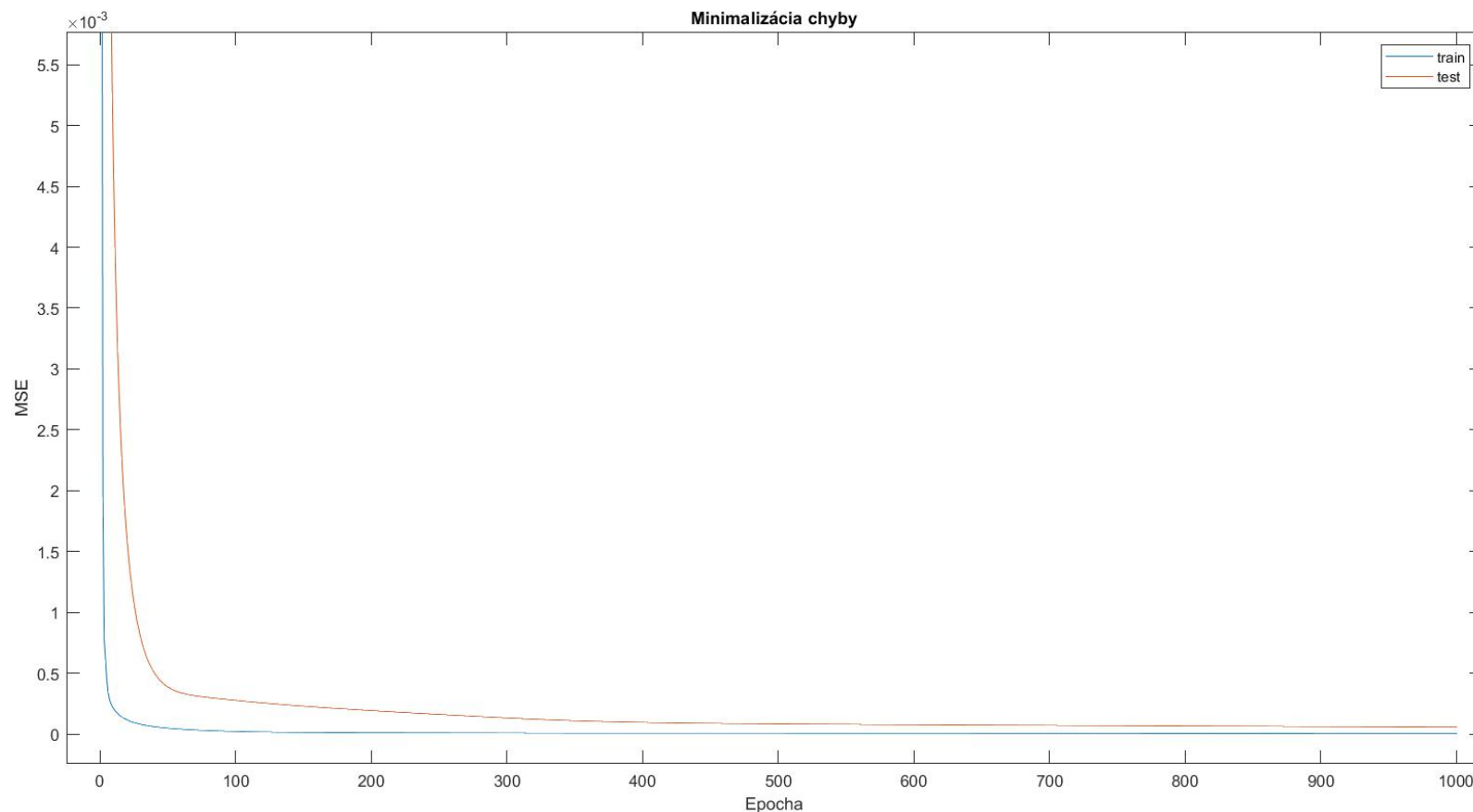
        IF "NeuralNetwork".Layer[#layers].TransferFunction = #logsig THEN
            "NeuralNetwork".Layer[#layers].Neural[#neural1].Value := 1 / (1 + (EXP(- (#intrinsicactivity))));
            "NeuralNetwork".Layer[#layers].Neural[#neural1].DerTranFun := EXP(#intrinsicactivity) / (EXP(2 *
            #intrinsicactivity) + (2 * EXP(#intrinsicactivity)) + 1);
        END_IF;
    END_FOR;
END_FOR;
```

...

Soft computing v riadiacich systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Trénovanie v PLC ...



Soft computing v radiacích systémoch

Neurónová sieť perceptrón pre PLC (Peter Štefaňák)

Porovnanie

Tabuľka 2. Výstupy z trénovacej sady

Číslo výstupu z učiacej sady	Žiadaný výstup	Výstup z Matlabu	Výstup z TIA Portalu
[1]	0.002752027	0.00975630230110	-0.000581293701171875
[2927]	1.953205	1.9631	1.953205

Tabuľka 3. Výstupy z testovacej sady

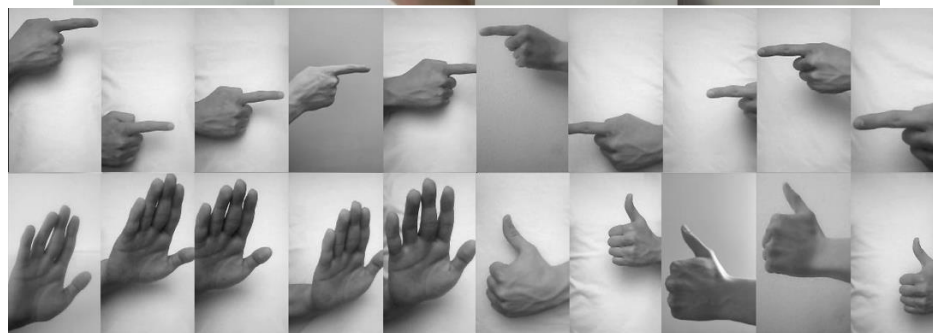
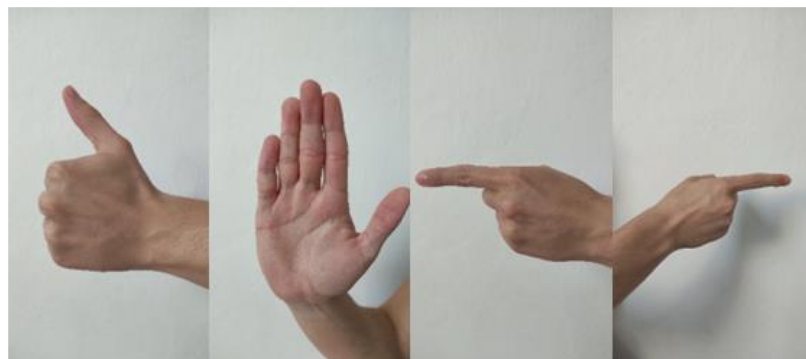
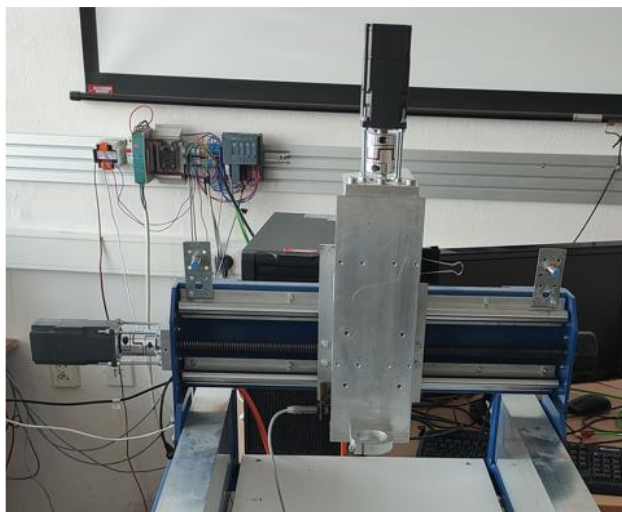
Číslo výstupu z testovacej sady	Žiadaný výstup	Výstup z Matlabu	Výstup z TIA Portalu
[30]	0.005132827	0.01033111211229	0.000836565184831619
[2874]	1.953199	1.963106662825297	1.953199012813

Soft computing v radiacich systémoch

Servo System Control Using Gesture – DP + článok (Peter Štefaňák)

Ciele:

- Databáza gest pre riadenie servo systému (model CNC)
- Implementácia CNN (bez trénovania) v PLC v jazyku ST
- Export z Matlabu



Soft computing v radiacich systémoch

Servo System Control Using Gesture – DP + článok (Peter Štefaňák)

Časť XML:

```
DATA BLOCK "CNN4"  
{ S7 Optimized Access := 'TRUE' }  
VERSION : 0.1  
NON RETAIN  
VAR  
    NumPairsLayers : Int;  
    InputLayer : Array[1..1] of Struct  
        Height : Int;  
        Width : Int;  
        Channel : Int;  
        Mean : Array[1..1] of Struct  
            Value : LReal;  
        END STRUCT;  
    END STRUCT;
```

Soft computing v radiacich systémoch

Servo System Control Using Gesture – DP + článok (Peter Štefaňák)

```
ConvolutionLayer : Array[1..3] of Struct
```

```
    OutputHeight : Int;
```

```
    OutputWidth : Int;
```

```
    Filter : Int;
```

```
    NumChannels : Int;
```

```
    NumFilters : Int;
```

```
    Stride : Int;
```

```
    PaddingUP : Int;
```

```
    PaddingDOWN : Int;
```

```
    PaddingLEFT : Int;
```

```
    PaddingRIGHT : Int;
```

```
    WeightsF : Array[1..6] of Struct
```

```
        WeightsCh : Array[1..6] of Struct
```

```
            rows : Array[1..3] of Struct
```

```
                columns : Array[1..3] of Struct
```

```
                    Value : LReal;
```

```
            END STRUCT;
```

```
        END STRUCT;
```

```
    END STRUCT;
```

```
END STRUCT;
```

Soft computing v radiacich systémoch

Servo System Control Using Gesture – DP + článok (Peter Štefaňák)

ConvolutionLayer : Array[1..3] of Struct

 OutputHeight : Int;

 OutputWidth : Int;

 Filter : Int;

 NumChannels : Int;

 NumFilters : Int;

 Stride : Int;

 PaddingUP : Int;

 PaddingDOWN : Int;

 PaddingLEFT : Int;

 PaddingRIGHT : Int;

 WeightsF : Array[1..6] of Struct

 WeightsCh : Array[1..6] of Struct

 rows : Array[1..3] of Struct

 columns : Array[1..3] of Struct

 Value : LReal;

 END STRUCT;

 END STRUCT;

 END STRUCT;

 END STRUCT;

Soft computing v radiacich systémoch

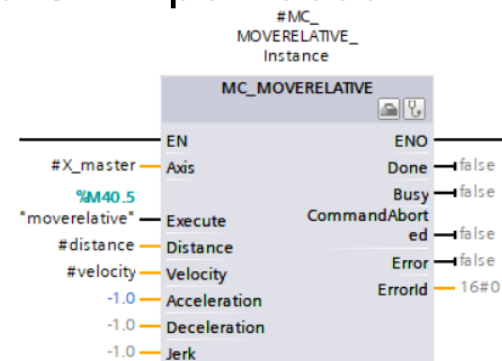
Servo System Control Using Gesture – DP + článok (Peter Štefaňák)

Problémy:

- Rýchlosť komunikácie obrazu cez OPC
- Veľkosť projektu (nutnosť nasadenia na Soft PLC - CPU 1515SP PC F + HMI)
- Rýchlosť CNN (cca 1.5s) => nepoužiteľné pre realtime riadenie

Algoritmus:

- Výber obr. uloženého v DB
- Výpočet výstupu CNN
- Ovládanie servomotorov podľa výstupu CNN pomocou MFB (MC Power, MC Halt, MC MoveVelocity, ...)



Soft computing v radiacich systémoch

Servo System Control Using Gesture – DP + článok (Peter Štefaňák)

TABLE I
CONFUSION MATRIX

	CNN outputs				
Gesture	Nothing	Start	Stop	Left	Right
Nothing	400	0	0	0	0
Start	0	393	1	5	1
Stop	0	4	396	0	0
Left	2	1	0	397	0
Right	0	3	0	0	397

TABLE II
COMPARISON OF THE CALCULATED CNN OUTPUTS, RECOGNIZED
GESTURES AND EVALUATION TIMES

Gesture	Application	CNN output	Classification	T [s]
Start	Matlab	0.9833	Start	0.1303
	PLC	0.9937	Start	1.3034
Stop	Matlab	1.0000	Stop	0.0134
	PLC	0.9999	Stop	1.3066
Nothing	Matlab	0.9712	Nothing	0.0571
	PLC	0.9890	Nothing	1.3377