

FUZZY SYSTEM FOR PLC

L. Körösi, D. Turcsek

Institute of Control and Industrial Informatics,

Slovak University of Technology, Faculty of Electrical Engineering and Information Technology

Abstract

Programmable Logic Controller (PLC) is a device used to solve complex control tasks in automation of electromechanical processes, such as control of machinery on factory assembly lines, etc. It is a user-programmable digital computer, which has compared to a standard computer some specific features. Functions in programmable logic controllers libraries are simple (bit operations, summations, subtractions, multiplications, divisions, etc.) or complex (sine, cosine, absolute value, vector summations, PID, etc.) mathematical functions but often without fuzzy systems, while PLC systems are currently the most commonly used control systems in industry. The aim of the proposed paper is to present a universal fuzzy system's design for PLC and the principle of Matlab fuzzy system conversion into PLC's fuzzy structure.

1 Introduction

Fuzzy logic was first proposed in by Lotfi A. Zadeh in 1965. Since then Fuzzy logic has emerged as a powerful technique for the controlling industrial processes, household and entertainment electronics, diagnosis systems and other expert systems. Rapid growth of this technology has actually started from Japan and then spread to the USA and Europe. Fuzzy logic is extremely useful in applications where we have a complex process and because of nonlinearities or time-varying responses, it is impossible to mathematically model the process. Often, traditional control methods such as PID control can't provide adequate control for these types of applications. Typically, these processes are still controllable by using and applying the expert knowledge of operators who have learned how the process responds to various input conditions [4]. The most common industrial control systems are Distributed Control System (DCS) and PLC's. DCS is a computerized control system used to control the production lines in the industry as oil refining plants, chemical plants, pharmaceutical manufacturing, etc. where continues control (PID loops) is dominating. PLC systems were typical for discrete (event) control – automotive, electronics, etc. Their primary goal was to replace the relay technology. Nowadays they have wide instruction libraries including function block for continues control (well-designed PID, lead-lag blocks, etc.) but there are missing libraries for intelligent control (fuzzy systems and neural networks). The proposed paper will summarize some existing fuzzy toolboxes for PLCs and present a universal fuzzy system for PLC with a methodology to convert Matlab fuzzy system into PLCs fuzzy structure.

2 Review of existing fuzzy toolboxes for PLC systems

RSLogix5000 Fuzzy Designer

RSLogix5000 Fuzzy Designer - software package from Allen - Bradley designed for the creation of fuzzy systems and hierarchical fuzzy systems. It is mainly used in the following applications:

- Industrial automation and control systems (controller, supervisor, process model).
- Process diagnostics and intelligent monitoring systems (process state classification).
- The process of decision-making and forecasting (decision support systems).
- Forecasting (prediction model).

Fuzzy Designer provides a function block-type environment to create your fuzzy logic algorithms. Simple point and click interfaces make it easy to define, for example, the fuzzification membership functions (MF) or the rule block definitions. Fuzzy Designer provides a set of built-in components which allows to easily building hierarchical fuzzy systems. Once fuzzy logic algorithm was created, Fuzzy Designer integrates it with Logix controllers by using the new add-on instruction feature in RSLogix 5000 software [2].

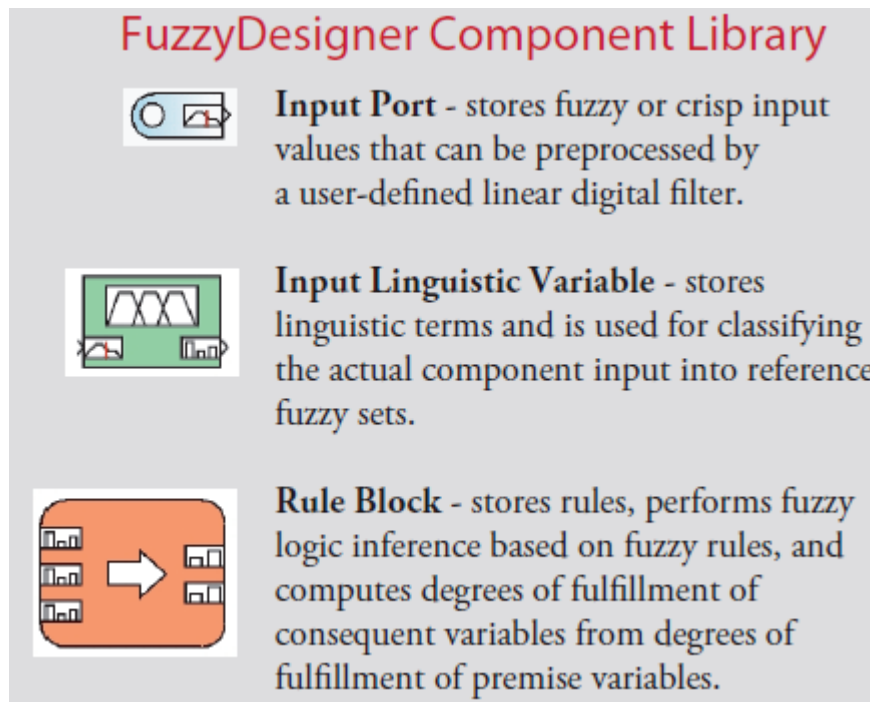


Figure 1: Examples of Fuzzy Designer library components

SIMATIC S7 Fuzzy Control

The S7 Fuzzy Control software package consists of two individual products:

- The product *Fuzzy Control* mainly contains the control block (function block - FB) and the data block (DB).
- The product *Configuration Fuzzy Control* contains the tool for configuring the control block.

The FB is already prepared in its full range of performance and with all algorithms for configuration and assigning parameters. A user-friendly tool is available for the configuration and parameter assignment of this function block (Fig.2). Fuzzy controllers are easy to configure on the basis of Fuzzy Control because their functionality is limited to the definition and execution of core functions in fuzzy theory. An instance data block in the CPU of the programmable controller forms the interface between the function block, the configuration tool, and the user. It's possible to download a number of fuzzy applications to a CPU and run them there. Each application is stored in a separate data block; the number of the data block can be freely assigned (Fig. 3) [3].

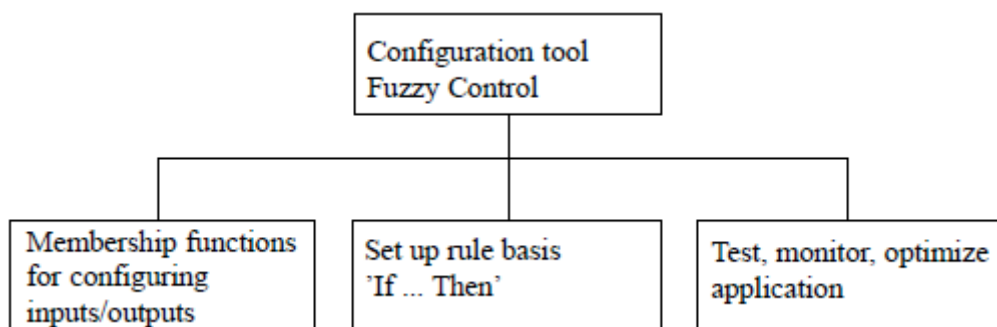


Figure 2: Block diagram of the configuration tool sub functions

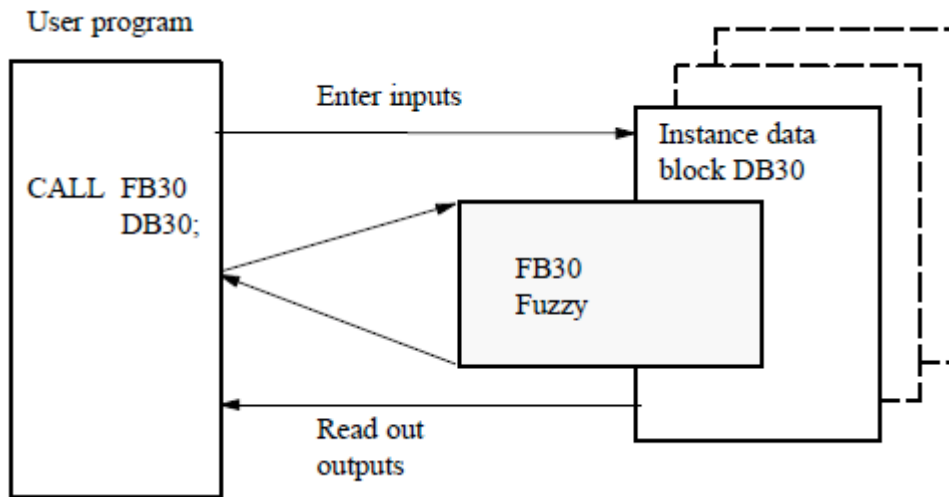


Figure 3: Structure of the block calls

3 Fuzzy system design for PLC

The fuzzy structure was designed in RsLogix5000 software used for ControlLogix and CompactLogix PLC's programming. RSLogix 5000 offers an easy to use IEC61131-3 compliant interface, symbolic programming with structures and arrays and a comprehensive instruction set that serves many types of applications. It provides ladder logic, structured text, function block diagram and sequential function chart editors for program development. Design of fuzzy system was implemented in ladder logic and structured text. Ladder logic is the most often used graphical programming language and the structured text which is similar to Pascal is suitable for iteration loops (repeat-until; while-do), conditional executions (if-then-else; case) and complex mathematical computations.

There are three main parts of the designed fuzzy structure: *fuzzification*, *inference mechanism* and *defuzzification*.

Fuzzification is the first step in the fuzzy inferencing process. This involves a domain transformation where crisp inputs are transformed into fuzzy inputs. Crisp inputs are exact inputs measured by sensors and passed into the control system for processing, such as temperature, pressure, rpm's, etc. Each crisp input that is to be processed by the FIU has its own group of membership functions or sets to which they are transformed. This group of membership functions exists within a universe of discourse that holds all relevant values that the crisp input can possess. The following figures (Fig. 4 and Fig. 5) show the designed structure of membership functions within a universe of discourse for a crisp input.

[-] input1			str_trmf[3]
+ input1[0]			str_trmf
+ input1[1]			str_trmf
+ input1[2]			str_trmf

Figure 4: Input1 structure (3 memberships functions are defined)

[-] input1[0]	(...)	(...)		str_trmf
input1[0].x1	0.0		Float	REAL
input1[0].x2	20.0		Float	REAL
input1[0].x3	40.0		Float	REAL
input1[0].in	2.3		Float	REAL
input1[0].out	0.114999995		Float	REAL

Figure 5: Example of triangular membership function tag (structure) definition in RsLogix5000

Each linguistic variable consists of three points x_1, x_2, x_3 , which unambiguously define the triangular membership function (Fig.6). Variables *in* and *out* define the crisp input and the membership functions output for the later processing.

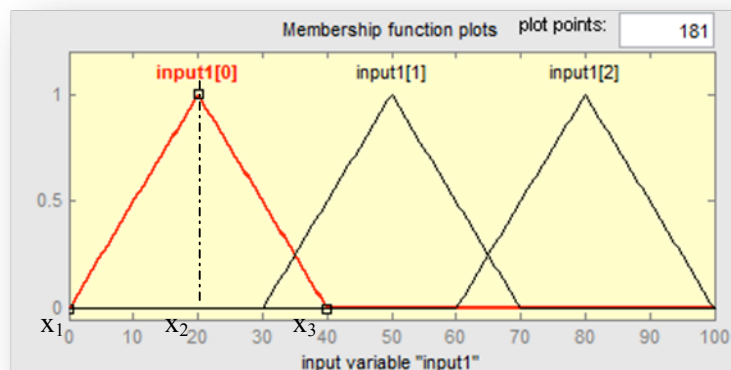


Figure 6: Linguistical variables (membership functions) -INPUT1 (variable input1[0]) in more detail

Example of PLC source code (in Structured Text language) for output computation is listed below:

```

if ((inputi[j].in >= inputi[j].x1) AND (inputi[j].in <= inputi[j].x2)) then
    a3 := -1; b3 := inputi[j].x2 - inputi[j].x1; c3 := inputi[j].x1; d3 := 0;
    e3 := -(a3 * inputi[j].x1 + b3 * d3); y3 := -(a3 * inputi[j].in + e3) / b3; inputi[j].out := y3;
elseif ((inputi[j].in > inputi[j].x2) AND (inputi[j].in <= inputi[j].x3)) then
    a3 := 1; b3 := inputi[j].x3 - inputi[j].x2; c3 := inputi[j].x2; d3 := 1;
    e3 := -(a3 * inputi[j].x2 + b3 * d3); y3 := -(a3 * inputi[j].in + e3) / b3; inputi[j].out := y3;
else
    inputi[j].out := 0;
end_if;

```

Inference mechanism is based on *IF ... THEN ...* rules by using well known MIN MAX methods (MIN for AND, MAX for OR). Each rule sets the *defuzz_val.in* values which are used for defuzzification process.

Example of PLC source code inference mechanism is listed below:

```

if (rule[30].out > defuzz_val.in1) then
    defuzz_val.in1 := rule[30].out;
else
    defuzz_val.in1 := defuzz_val.in1;
end_if;

```

Defuzzification is the process of producing a quantifiable result in fuzzy logic, given fuzzy sets and corresponding membership degrees. A common and useful defuzzification technique is center of gravity (centroid). This part of the proposed algorithm uses values from inference mechanism for all rules and calculates the *x* coordinate of the centroid which is the defuzzified value.

Creating a fuzzy system can be made manually or converting existing fuzzy structure from Matlab. The Matlab fuzzy structure is stored in text file. Examples of a membership functions is below:

```
MF1='input1[0]':'trimf',[0 20 40]
MF2='input1[1]':'trimf',[30 50 70]
MF3='input1[2]':'trimf',[60 80 100]
```

The structure is exported into XML format (*.L5K file) compatible with Logix systems (RsLogix5000 projects). Example of the conversion for the MF1, MF2 and MF3:

```
input1 : str_trimf[3] (Description := " input1 – trimf functions definition") := [[0.00000000e+000,
0.00000000e+001, 0.00000000e+001, 0.00000000e+000, 0.00000000e+000], [0.00000000e+001,
0.00000000e+001, 0.00000000e+001, 0.00000000e+000, 0.00000000e+000] ,[0.00000000e+001 ,
0.00000000e+001, 0.00000000e+002, 0.00000000e+000, 0.00000000e+000]];
```

4 Case study and simulation results

To verify the functionality of the designed fuzzy system, we generated fuzzy results of the fuzzy model in the PLC and compared results from Matlab fuzzy system.

Sample (SET1) of input values in Matlab and crisp value using centroid defuzzification.

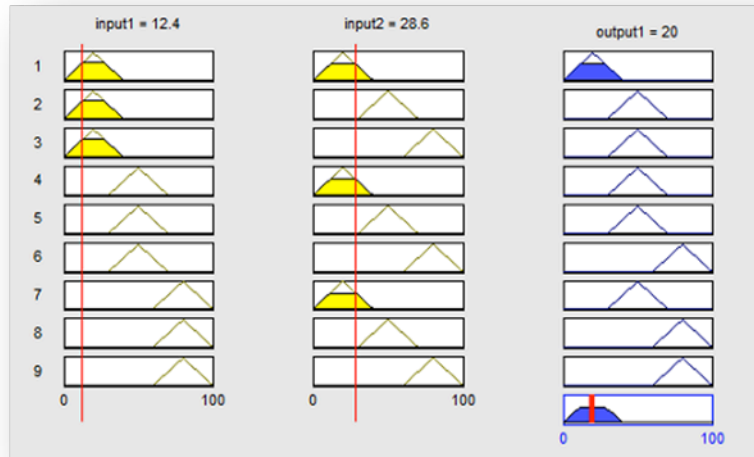


Figure 7: Matlab Fuzzy Rule Editor

RSLogix5000 - set of input values and the output defuzzification:

input1[0]	{ . . . }	{ . . . }		str_trimf
input1[0].x1	0.0		Float	REAL
input1[0].x2	20.0		Float	REAL
input1[0].x3	40.0		Float	REAL
input1[0].in	12.4		Float	REAL
input1[0].out	0.62		Float	REAL

Figure 8: Input1 values

input2[0]	(...)	(...)		str_trimf
input2[0].x1	0.0		Float	REAL
input2[0].x2	20.0		Float	REAL
input2[0].x3	40.0		Float	REAL
input2[0].in	28.6		Float	REAL
input2[0].out	0.57		Float	REAL

Figure 9: Input2 values

defuzz_val	(...)	(...)		str_defuzz_in
defuzz_val.in1	0.57		Float	REAL
defuzz_val.in2	0.0		Float	REAL
defuzz_val.in3	0.0		Float	REAL

Figure 10: Prepared values for defuzzification

output_defuzz	20.00001		Float	REAL
---------------	----------	--	-------	------

Figure 11: Defuzzified value using centroid

Sample (SET2) of input values in Matlab and crisp value using centroid defuzzification.

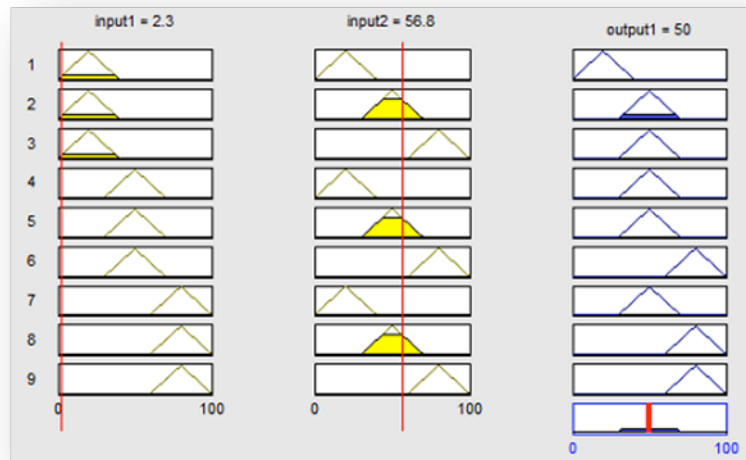


Figure 12: Matlab Fuzzy Rule Editor

input1[0]	(...)	(...)		str_trimf
input1[0].x1	0.0		Float	REAL
input1[0].x2	20.0		Float	REAL
input1[0].x3	40.0		Float	REAL
input1[0].in	2.3		Float	REAL
input1[0].out	0.114999995		Float	REAL

Figure 13: Input1 values

input2[1]	{...}	{...}		str_trimf
input2[1].x1	30.0		Float	REAL
input2[1].x2	50.0		Float	REAL
input2[1].x3	70.0		Float	REAL
input2[1].in	56.8		Float	REAL
input2[1].out	0.66		Float	REAL

Figure 14: Input2 values

defuzz_val	{...}	{...}		str_defuzz_in
defuzz_val.in1	0.0		Float	REAL
defuzz_val.in2	0.114999995		Float	REAL
defuzz_val.in3	0.0		Float	REAL

Figure 15: Prepared values for defuzzification

output_defuzz	49.999905		Float	REAL
---------------	-----------	--	-------	------

Figure 16 Defuzzified value using centroid

5 Conclusion

The proposed paper presented a fuzzy system design for PLC system and the automatic fuzzy structure conversion into PLC. The fuzzy toolbox has been verified on several examples and it's suitable for modeling and control nonlinear processes. The fuzzy system can be designed directly in Matlab and after sets of simulations the final fuzzy system can be deployed into PLC without manual configurations. The extension of the fuzzy system with new fuzzy MF's and defuzzification methods is planned in the future.

ACKNOWLEDGMENT

This paper was supported by the Slovak Scientific Grant Agency VEGA, Grant no. 1/1105/12, and WEGA No. 032STU-4/2011.

References

- [1] The MathWorks, Inc., 1994 – 2011, Available on the Internet:
<http://www.mathworks.com/products/fuzzy-logic/description3.html>
- [2] Publication LOGIX – UM004A – EN – P – March 2007, Rockwell Automation, Inc. *RSLogix5000 FuzzyDesigner*, U.S.A., 2007. Available on the Internet:
http://literature.rockwellautomation.com/idc/groups/literature/documents/um/logix-um004_-en-p.pdf
- [3] Siemens AG, 1998. *Simatic S7 Fuzzy Control*, Germany, 1998. Available on the Internet:
http://cache.automation.siemens.com/dnl/TU1Nzg1NwAA_1135654_HB/Fuzzy_e.pdf

[4] Fuzzy rule-based systems. Available on Internet:
<http://www.data-machine.nl/fuzzy1.htm>

Ladislav Körösi, Dávid Turcsek

Institute of Control and Industrial Informatics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology,
Ilkovičova 3,
812 19 Bratislava, Slovak Republic
e-mail: ladislav.korosi@stuba.sk, xturcsek@is.stuba.sk