Fuzzy Logic Control on FPGA for Solar Tracking System

Ricardo Antonio-Mendez, Jesus de la Cruz-Alejo and Ollin Peñaloza-Mejia

Abstract This paper proposes a control technique for solar tracking system through the processing of fuzzy logic implemented on FPGA. The control is designed by calculating and storing membership values in lookup tables as integer values which are addressed in all stages involved in the fuzzy logic system according to Mamdani rules and max—min implication model. A defuzzification method using alpha-levels is proposed. Results are presented to validate the theoretical analysis.

Keywords Fuzzy logic · Alfa-levels · Sun tracking · FPGA · Mamdani

1 Introduction

Currently one of the mathematical disciplines with the highest number of followers is the fuzzy logic technique, which is the logic that uses expressions that are neither completely true nor completely false, i.e., is the logic applied to concepts that can take a any truth value in a set of values ranging between two extremes, absolute truth and complete falsity. Central to those based on the theory of fuzzy logic systems is that, unlike those based on classical logic, have the ability to acceptably reproduce the usual modes of reasoning, considering the certainty of a proposition is a matter of degree [1, 2]. Thus, the most attractive features of fuzzy logic are its

R. Antonio-Mendez (✉) · J. de la Cruz-Alejo · O. Peñaloza-Mejia
Mechatronic Department, Tecnológico de Estudios Superiores de Ecatepec, Ecatepec, Estado de México, Mexico
e-mail: ric_xof@hotmail.com

J. de la Cruz-Alejo
e-mail: jesus_ch517@hotmail.com

O. Peñaloza-Mejia
e-mail: openaloza@tese.edu.mx

M. Ceccarelli and E.E. Hernández Martinez (eds.), *Multibody Mechatronic Systems*, Mechanisms and Machine Science 25, DOI 10.1007/978-3-319-09858-6_2

flexibility, tolerance for imprecision, its ability to model non-linear problems, and its basis in natural language.

In the other hand, implementation can be using general-purpose processors and depending fully on software in the realization of the system or adapting a general-purpose processor to perform dedicated fuzzy instructions. The approach is a trade-off between speed and generality. An alternative is using an exclusive hardware to perform the fuzzy operations as a closely related approach, through dedicated fuzzy circuits or Application Specific Integrated Circuits (ASICs). The approach leads to relatively high-speed operation, but is more costly. FPGAs are hardware devices used as user-programmable ASICs. The availability of software tools to generate efficient and flexible hardware description configurations automatically also brings easiness to the reconfiguration process. Moreover, FPGA designs can already be modeled, simulated and verified.

For the case of solar-concentrating collectors, some form of tracking mechanism is usually employed to enable the collector to follow the sun. This is done by monitoring the variation in degrees. Monitoring tracking systems can be classified based on their movement. This can be a single axis or two axes. In the case of a single-axis mode, the motion can be in various ways: east-west, north-south, or parallel to the earth's axis [2—4].

The paper describes the fastest way to program the FPGA based on the corresponding application program, which in this case is the code for the sun tracking control. Section 2 characteristics of the sensors and actuators, as well as communication protocols implemented on the FPGA program for each device are described. Section 3, experimental results are presented. Section 4, a comparison between defuzzification method used in the paper against centroid method is presented.

2 Fuzzy Logic System

Figures 1, 2 and 3 shows the block diagram of the overall system. Data acquisition is carried out by four brightness sensors which convert light intensity into digital output values. The output values are stored in internal registers which are accessed via the I^2C communication protocol. It serves as interface for input values to the FLC. Outputs values of the FLC are the control actions which are communicated to the servomotors through the RS-232 protocol in digital mode, which contain the commands for positioning, according to the own configuration of each one servomotor. Hardware description, processing, setup and characteristics of the devices used in the system are presented next.

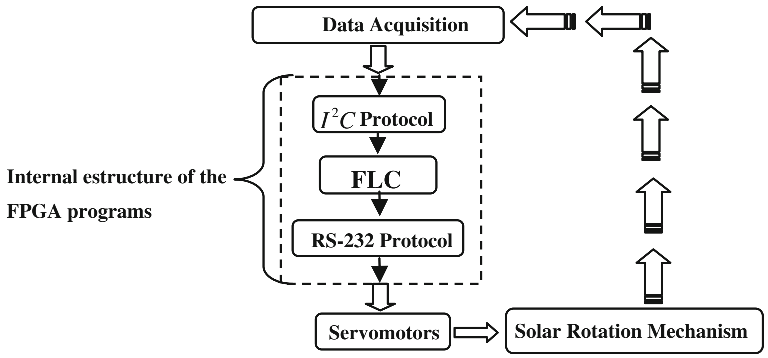


Fig. 1 Block diagram for the sun tracking system

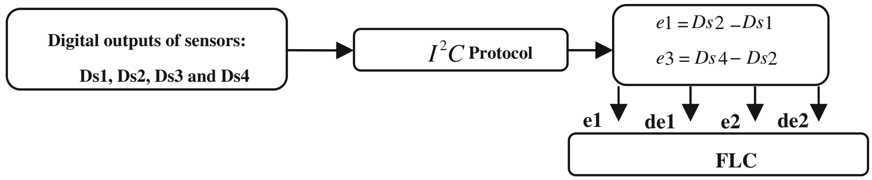


Fig. 2 Data acquisition for the FLC system

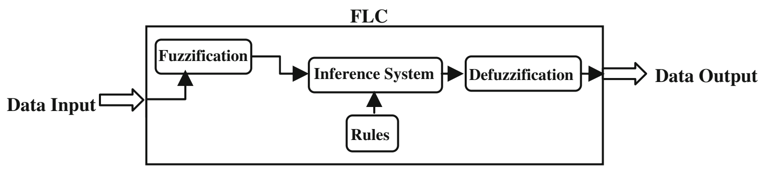


Fig. 3 Fuzzy logic control (FLC)

2.1 Fuzzification

The membership function chosen for this control was triangular, which is given by the parameters a, b, c as follows:

$$Triangulo(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \tag{1}$$

Using max—min operators, (1) can be represented as:

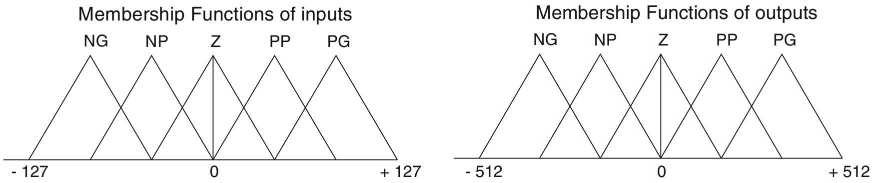


Fig. 4 Membership function for inputs and outputs

$$triangulo(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (2)$$

Using (2), the membership values for each input or output variable are obtained. For fuzzification stage, alpha-levels method was chosen. Input variables were expressed by five linguistic variables positive big (PB), positive small (PS), zero (ZO), negative small (NS) and negative big (NB). Membership functions of the fuzzy system are shown in Fig. 4. The universe of discussion for each input and output variable was expressed by five linguistic variables positive big (PB), positive small (PS), zero (ZO), negative small (NS) and negative big (NB).

The membership values are obtained through the programming of (2) in a spreadsheet on a pc. The resultants values are captured in look up tables which are stored in the internal program in order to be accessed to them in subsequent stages of the fuzzy logic system control. For example: Given an input value x, the membership value obtained is a floating-point value in a range between 0 and 1. This kind of value is complicated to implement in hardware, besides that involves the use of more resources on the device. So, the values are scaled depending on the resolution required, according to:

$$\left[\mu(x) = Eq.(2)_{floathing\ point} \times 2^n - 1\right] = \mu(x)_{integer\ value} \quad (3)$$

This manner, using (3), floating point value is scaled to a $2^n - 1$ resolution, where n represent the number of bits used in the system. The code to obtain all membership values for negative_big set is as follows:

type Negative_Big is array (a to c) of integer range 0 to $2^n - 1$.
constant f_m1: Negative_Big: = (a => 0,1 => 6,2 => 12,4 => 18,….,c => $2^n - 1$*);*
for i in a to *c loop*
if (*x* = *i) then*
μ(x) = f_m1(i)
end if;
end loop;

where *a* and *c* are the parameters of the membership function given by (1) and (2).

In this case, because the sensors have a resolution of 8 bits($2^n - 1$), the input universe of discussion is partitioned from −127 to 127 as is shown in Fig. 4. A

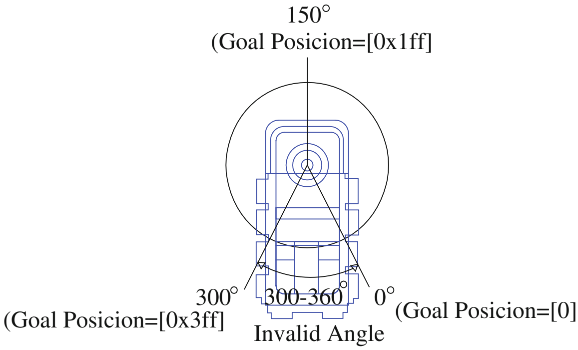


Fig. 5 Range of movement of the servomotors

value of 0 indicates the lowest value of lighting and a value of 255 highest level. This indicates that the entry for the maximum value $e1 = Ds2 - Ds1$either negative or positive is $2^n - 1$. Depending on whether *Ds2* is greater than *Ds1,* the result is a positive number or otherwise *Ds1* is greater than $Ds2$, which are negatives values. The universe from −512 to 512 indicated in Fig. 4 for the output of the system is proposed due to the movement of the servomotors have 1024 possible values. So, 1024 represents a position of 300°, 512 represents the midpoint, as shown in Fig. 5.

2.2 Rules

In this work, Mamdani's method is used. In fuzzy logic controller a descriptive verbal rules (If—Then rules) are used to describe the relation among inputs and outputs, according to:

$$R^{(l)} : IF\ x_1\ is\ F_1^l\ and\ldots and\ x_n\ is\ F_n^l\ THEN\ y\ is\ G^l \tag{4}$$

where:

$$(x_1\ldots x_n)\ Represent\ the\ input\ variable$$
$$(y)\quad Represent\ the\ output\ variable$$
$$(F\ y\ G)\ Represent\ the\ membership\ function\ of\ fuzzy\ set.$$

So, rules are activated according to (4). Table 1 shows the corresponding rules for the fuzzy logic system.

Table 1 Corresponding rules for the fuzzy logic system

			e1			
		NG	NP	Z	PP	PG
	NG	NG	NP	NP	NP	NP
	NP	NP	Z	PP	NP	Z
de1	Z	NP	PP	Z	PG	PP
	PP	Z	PP	PP	Z	PP
	PG	PP	PP	PP	PP	PG

2.3 Inference

The inference method chosen for this work is given by:

$$\mu_{A \to B}(x, y) = \min[\mu_A(x), \mu_B(y)] \quad (5)$$

It involves the comparison between two integer values, selecting the minimum value to activate rules. Now, in order to programming the fuzzy system, two arrays of rows and columns numbers which contain the maximum membership functions are programming. In these arrays are store the membership values which, corresponding to the fuzzy sets where they will be kept, i.e., if an input value falls within one or two fuzzy sets, the corresponding membership values for each one will be different in each set, therefore they will be stored in the array. Figure 6 shows the flow chart for the search and storing for all values in the array. The value *Data_inx* having *e1,* is stored in this variable and *a* and *c* values corresponding to the parameters of (1) (triangular membership function). These are the ranges in which the input value is looked *Data_inx*.

Whenever *Data_inx* has a constant value that matches the value previously stored in the tables corresponding to the membership function in question, this will be stored in the array *Wnx*. Table 2 shows vectors obtained from the search processes for *Data_inx* (*e1*) and *Data_iny* variable (*de1*).

Taking into account the resulting vectors from the search, the inference process shown in Fig. 7, is performed. Inference process results are given in Table 3, as can be seen only the array of rules denominated $Wn\_rules(i,j)$, are stored. This array has the same dimension as Table 1 have. This means that the input values which activate output linguistic rules are stored in the array $Wn\_rules(i,j)$.

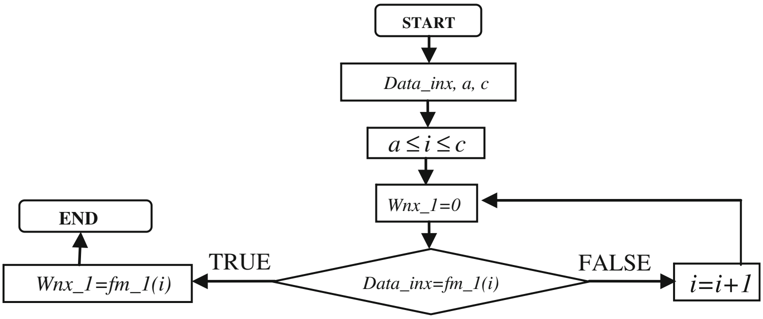


Fig. 6 Flow chart for the search process

Table 2 For data *inx* and data *iny*

Digital inputs	*Vectors*	*fm_1*	*fm_2*	*fm_3*	*fm_4*	*fm_5*
Data_inx	*Wnx=*	*Wnx_1*	*Wnx_2*	*Wnx_3*	*Wnx_4*	*Wnx_5*
Data_iny	*Wny=*	*Wny_1*	*Wny_2*	*Wny_3*	*Wny_4*	*Wny_5*

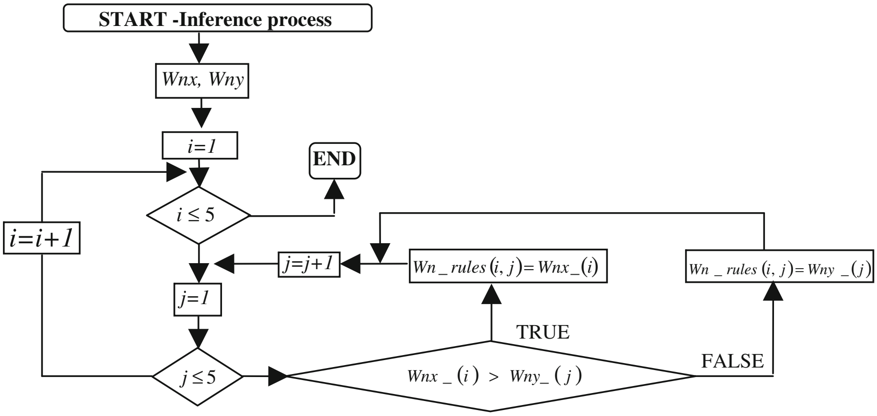


Fig. 7 Flow chart for the inference process

2.4 Aggregation

Aggregation stage is done by:

$$\mu_{B^l}(y) = \max\left[\mu_{F_1^l}(x_1), \ldots, \mu_{F_n^l}(x_n)\right] \quad (6)$$

It shows the global union of fuzzy sets and rules activated. It takes into account only those rules that have a non-zero value. The aggregation process is carried out by the method of maximum Eq. (6), for each one enabled rule whose value is

Table 3 Vector obtained by the Inference process

				e1		
		NG	NP	Z	PP	PG
	NG	*Wn_rules (1,1)*	*Wn_rules (1,2)*	*Wn_rules (1,3)*	*Wn_rules (1,4)*	*Wn_rules (1,5)*
	NP	*Wn_rules (2,1)*	*Wn_rules (2,2)*	*Wn_rules (2,3)*	*Wn_rules (2,4)*	*Wn_rules (2,5)*
de1	Z	*Wn_rules (3,1)*	*Wn_rules (3,2)*	*Wn_rules (3,3)*	*Wn_rules (3,4)*	*Wn_rules (3,5)*
	PP	*Wn_rules (4,1)*	*Wn_rules (4,2)*	*Wn_rules (4,3)*	*Wn_rules (4,4)*	*Wn_rules (4,5)*
	PG	*Wn_rules (5,1)*	*Wn_rules (5,2)*	*Wn_rules (5,3)*	*Wn_rules (5,4)*	*Wn_rules (5,5)*

Table 4 Process for the aggregation process

				e1		
		NG	NP	Z	PP	PG
	NG	Wn_rules(1,1)	Wn_rules(1,2)	Wn_rules(1,3)	Wn_rules(1,4)	Wn_rules(1,5)
		↓	↓	↓	↓	↓
	NP	Wn_rules(2,1)	Wn_rules(2,2)	Wn_rules(2,3)	Wn_rules(2,4)	Wn_rules(2,5)
		↓	↓	↓	↓	↓
de1	Z	Wn_rules(3,1)	Wn_rules(3,2)	Wn_rules(3,3)	Wn_rules(3,4)	Wn_rules(3,5)
		↓	↓	↓	↓	↓
	PP	Wn_rules(4,1)	Wn_rules(4,2)	Wn_rules(4,3)	Wn_rules(4,4)	Wn_rules(4,5)
		↓	↓	↓	↓	↓
	PG	Wn_rules(5,1)	Wn_rules(5,2)	Wn_rules(5,3)	Wn_rules(5,4)	Wn_rules(5,5)

different of 0. Then, it is compared with another to know which one is the maximum, and so on as is shown in Table 4. It shows the way and how the maximum is saved by comparing the vector $Wn\_rules(i,j)$. In this case, e.g., taking NG column, membership function values are compared following, (1,1,), (2,1), (3,1), (4,1) and (5,1). So, the maximum value between them is the result that activates the output. It is the value that was used in the alpha-levels defuzzification (αi) process, illustrated in Fig. 8, where represents the membership value as a result for the aggregation.

2.5 Defuzzification

The method chosen for defuzzification is the Center of slice Area Average (COSAA) [1], which is given by:

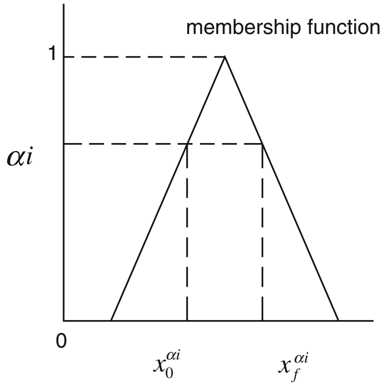


Fig. 8 Parameters of membership function

$$COSAA = \frac{\sum_{i=0}^{\alpha\max}\left(\frac{x_f^{\alpha i} - x_0^{\alpha i}}{2}\right) + x_0^{\alpha i}}{\alpha\max} \tag{7}$$

where $x_0^{\alpha i}$ and $x_f^{\alpha i}$ are parameters given according to the Fig. 8.

2.6 Experimental and Simulated Results

Figure 9a, b show the position of the sensors in the system, where we can see the position of the sensors and the direction of the movement that each motor have in the FLC system. Figure 10 shows a comparison between the implemented method proposed and the MatLab toolkit. The toolkit of Matlab as Bisector and Centroid establish a non-significant difference for the control. The inputs to the controller are:

$$e1 = Ds2(k) - Ds1(k),\ and\ de1 = e1(k) - e1(k-1) \tag{8}$$

where *k,* is the sampling time. So, the process adapts the characteristics of the nonlinear behavior of the solar tracking. This means that either cloudy days or sunny days, the system adapts to the minimum or maximum conditions of solar radiation or in this case lighting. The novelty of this system is the reduction of operations performed on the FPGA. Therefore it only uses basic operations such as addition, subtraction, divisions and comparisons. In Table 5, the experimental output results values for the implemented proposed method and the MatLab are given. The discrepancy between them is less than an 5 % for all parameters over a full range contained in the universe of discussion X.

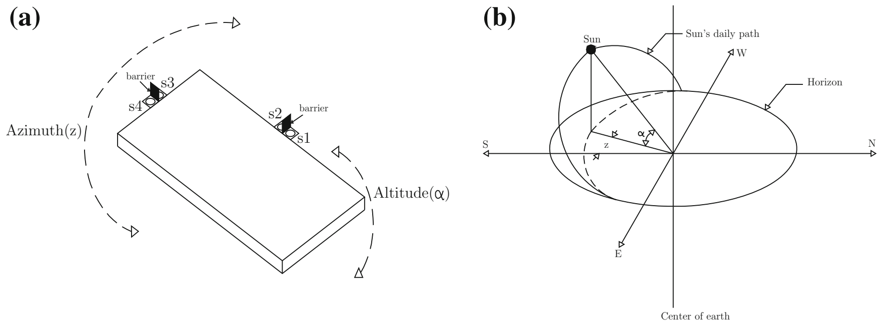
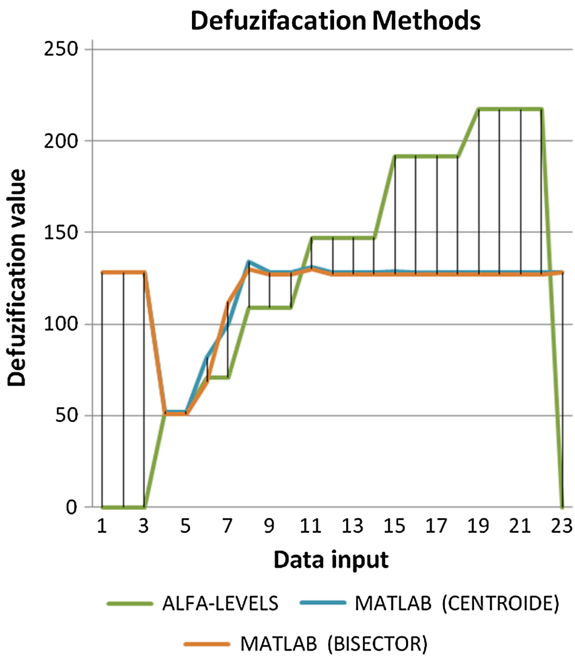


Fig. 9 **a** Motion Overviews, **b** daily path of the sun across the sky from sunrise to sunset

Fig. 10 Comparison between the system proposed and Matlab defuzzification methods



3 Conclusions

The proposed FPGA sun tracking control, based on fuzzy logic technique was implemented. It focuses on the optimization of hardware resources, conducting operations to obtain the values of membership functions for either input or output, accessing them as constant values using look-up tables. All the simulation and experimental results demonstrate the effectiveness of the FLC system. Through the comparison between them, the proposed system achieves a control performance and

Table 5 Comparative methods defuzification

Comparative table				
Inputs value		Outputs value (Z)		
X	Y	Methods of defuzification		
		Alfa-levels (proposed method)	MATLAB (centroide)	MATLAB (bisector)
0	0	0	128.0	128
14	14	0	128.0	128
14	20	0	128.0	128
25	30	52	52.0	51
35	40	52	52.0	51
50	65	71	81.7	68.8
70	80	71	99.8	112
85	95	109	134	130
100	110	109	128	127
115	120	109	128	127
245	255	0	128	128

allowing a faster and precise control of the sun tracking. Besides the proposed fuzzification method (COSAA), does not imply higher processing calculations. Moreover the devices used both data acquisition and those performing the control actions have a good behavior.

References

1. Zavala AH (2009) Arquitectura de Alto Rendimiento para sistemas difusos, Tesis, IPN, Enero
2. Chekired F, Larbes C, Mellit A (2012) Comparative study between two intelligent MPPT-controllers implemented on FPGA: application for photovoltaic systems. Int J Sustain Energ. doi:10.1080/14786451.2012.742896
3. Othman AM et al, Real world maximum power point tracking simulation of PV system based on Fuzzy Logic control. NRIAG J Astron Geophys 1(2):186–194
4. Nader Barsoum Curtin University of Technology, Sarawak, Malaysia, Implementation of a prototype for a traditional solar tracking system, UKSim 3rd European modelling symposium on computer modelling and simulation, Athens, Greece, 25 November 2009, pp 23–30

http://www.springer.com/978-3-319-09857-9

Multibody Mechatronic Systems
Proceedings of the MUSME Conference held in Huatulco, Mexico, October 21-24, 2014
Ceccarelli, M.; Hernandez, E.E. (Eds.)
2015, XIV, 581 p. 360 illus., Hardcover
ISBN: 978-3-319-09857-9