



HAL
open science

Efficient Design and Implementation of a Multivariate Takagi-Sugeno Fuzzy Controller on an FPGA

Abiel Aguilar, Madaín Pérez, Jorge L Camas, Héctor R Hernández, Carlos Ríos

► **To cite this version:**

Abiel Aguilar, Madaín Pérez, Jorge L Camas, Héctor R Hernández, Carlos Ríos. Efficient Design and Implementation of a Multivariate Takagi-Sugeno Fuzzy Controller on an FPGA. 2014 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE), Nov 2014, Cuernavaca, Mexico. pp.152 - 157, 10.1109/ICMEAE.2014.8 . hal-01627290

HAL Id: hal-01627290

<https://hal.science/hal-01627290>

Submitted on 2 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Efficient Design and Implementation of a Multivariate Takagi-Sugeno Fuzzy Controller on an FPGA

Abiel Aguilar-González, Madaín Pérez-Patricio,
Jorge L. Camas-Anzuetto, Héctor R. Hernández-De león, Carlos Ríos-Rojas
*Laboratorio de Optomecatrónica
Instituto Tecnológico de Tuxtla Gutiérrez
Tuxtla Gutiérrez, México*
13270869@ittuxtlagutierrez.edu.mx, mperez@ittg.edu.mx, jcamas@ittg.edu.mx,
hhernandezd@ittg.edu.mx, crios@ittg.edu.mx

Abstract—This article describes the design and efficient implementation of a Takagi Sugeno multivariable Fuzzy Logic Controller. The application selected is a temperature and humidity controller for a chicken incubator. This design was elaborated using VHDL applying intermediate simulations in order to check for functional verification of all modules integrating the controller. The created circuit was implemented on FPGA Cyclone II EP2C35F672C6 assembled in breadboard Altera DE2. Precision of the model on FPGA was calculated by contrasting the obtained results with the results of simulation using MATLAB & Fuzzy Logic Tool Box.

Keywords-fpga: vhdl; fuzzy

I. INTRODUCTION

During recent years the increment of applications for the fuzzy logic increased the necessity to find efficient forms for its application. For that purpose, controllers moved into VLSI (Very Large Scale Integration) circuits due to the implementation of highly complex circuits in these Chips. Recently, the high costs incurred in manufacturing VLSI chips in case of any modifications or redesign led to the use of Field Programmable Gate Array FPGA's [1]-[6].

Main advantages in the use of FPGA devices are: Simplicity to redesign the controllers based on specifications without incurring in Non-Recurring Engineering and the Parallel Processing capacity, obtaining real-time processing. According to the previously mentioned advantages, diverse authors have been motivated to study the efficient implementation of fuzzy controllers in HDL (Hardware Description Language) being simplified Floating Point Operations and reducing the number of products and quotients. In most of these investigations, authors determined the precision of the system comparing the output implemented in hardware with output simulated in MATLAB & Fuzzy Logic Tool Box. In [7] the implementation of a Mamdani controller using VHDL (Very High Speed Integrated Circuits Hardware Description Language) encoding and modular programming is described, obtaining an error of 0,7% in the output. The

implementation of a Mamdani controller on an FPGA is described in [8], in this paper it is proposed the control of a multiple input system managing to obtain an error of 0,8% at the output. It was demonstrated in [9] the implementation of a nonlinear controller on an FPGA, for this purpose it was used a recurrent fuzzy controller.

Due to the high consumption of hardware resources that fuzzy systems present in the stage of defuzzification, research has been made, analyzing the possibility of reducing hardware costs. A new concept of weighted average was used as exposed in [10], in which the most significant bits of calculation input are used, with this, it is managed to maintain a low consumption of resources, even though the input possesses highly numerical values. The controller proposed in [8] carries out the output's defuzzification using the centroid method:

$$\text{COG} = \frac{\int f(x)xdx}{\int f(x)dx}$$

where COG is the real value of the centroid and $f(x)$ is the function which defines the implication of the rules and x is the consequence for each rule, due to high costs in hardware planted in the operation, authors made an adjustment of the proposed algorithm in [11] consequently being capable to realise the process of division with low consumption of resources. Finally, a hybrid controller is proposed in [12], in which two fuzzy parallel controllers linked together by a single defuzzification stage, managing to reduce defuzzification costs to half of its total.

In this paper it's studied the efficient implementation of fuzzy controllers in HDL. New strategies appear to reduce costs as well as a comparison of hardware resources consumption and the accuracy of the system using different defuzzification methods. The application selected during this research is a temperature and humidity controller for a chicken incubator.

Nowadays the incubation plant is an unquestionable part of the chain of poultry production. Vitality or quality of the chickens is accepted as a critical parameter taking advantage of all genetic potential the chickens have. Incubation temperature is the main point of discussion, it's well known that the ideal temperature for this is 38 °C and an overheat causing several damage of the embryo . It is not known generally that temperatures below this value also affect the results. On the other hand, to maintain humidity between 85% and 87% during incubation is vital, in order to hatch a healthful brood satisfactorily.

II. CONTROLLER DESIGN

Knowing the design requirements it is possible to conceptualize a fuzzy controller. Two variables, *temperature* and *humidity* will be controlled. Two inputs will be needed: the error between the measured temperature and the expected temperature; the error between the measured humidity and the expected humidity; and three outputs were in charge to warm up, to cool up and to dampen respectively, see Figure 1



Figure 1: Black box diagram of the proposed controller.

Nowadays diverse fuzzy controllers exist. The Mamdani controller class is widely used since its inference motor is intuitively similar to human reasoning. Nevertheless, this type of controller implies a high computer load. On the other hand, a Sugeno controller type is computationally more efficient and works well with optimization and adaptative techniques, for this reason it becomes more appealing to use in control problems solving. A most significant difference between these controllers is the form in which the real output is generated. Mamdani employs a slow defuzzification process whereas Sugeno uses simpler numeric methods. It was decided to use a Sugeno controller type due to its high computer efficiency.

A. Input Functions

Following elements $\Delta temp$ and Δhum state the arithmetical difference between values measured by the sensors and expected values of temperature and humidity. Later on a mapping is presented in which it is considered the maximum clause values between the difference of measured values

and expected values equal to $\pm 30^\circ\text{C}$ in temperature and $\pm 30\%$ in humidity, considering expected humidity is equal to 86%. This values are appropriate in accordance with the ideal characteristics of temperature and humidity for chicken incubator.

If integer numbers are used and the proposed mapping is applied in Equation 1, as *input* are values for temperature or humidity at a moment of time *t*, its possible to conclude error system inputs $error_{(temp)}$ or $error_{(hum)}$ will always be integer numbers from 7 to 247, for this reason is possible to establish a universe of discourse of input function with values between 0 and 255. This adjustment will be very useful when implementing the system in HDL, since it allows to work with binary chains of 8 bits, obtaining a low consumption of hardware resources. Figure 2 shows the functions of membership for *temperature* variable. Figure 3 shows the functions of membership for *humidity* variable, its been used a trapezoidal function for the errors near zero, when the error is the zero variable it can use values from 126 to 128 that is from 85% to 87% thus fulfilling with the requirements design.

$$error_{(input)} = (\Delta_{(input)} * 4) + 127 \quad (1)$$

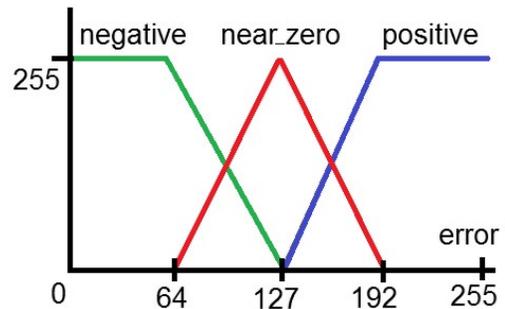


Figure 2: Membership functions for temperature.

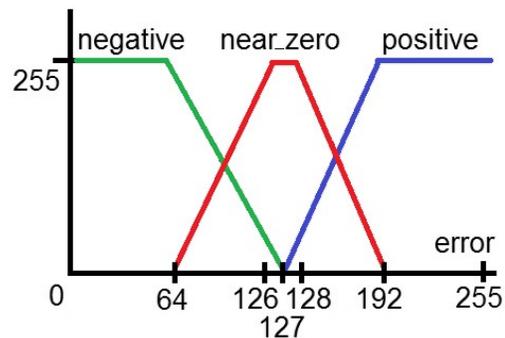


Figure 3: Membership functions for humidity.

B. Output Functions

The proposed system has three outputs, all of them can be modeled using Equations 2 and 3, as y_0, y_1 are real output values and x represents the implication value of the rules. Linear and pending equations with binary base have been used in order to reduce the complexity of the calculation, diminishing hardware resources demand.

$$y_0 = -8x + 255 \quad (2)$$

$$y_1 = x \quad (3)$$

C. Fuzzy Rules

Considering Equation 2 corresponds to points near zero to warm up, cool up and dampen, expressed by the variables Z_1, Z_2 and Z_3 respectively, Equation 3 takes values corresponding to the midpoints and maximum. A logical way to relate the system inputs to these equations is shown:

- IF $\text{error}_{(\text{temp})}$ IS positive THEN $Z_1 = 0, Z_2 = y_1$
- IF $\text{error}_{(\text{temp})}$ IS near_zero THEN $Z_1 = y_0, Z_2 = y_0$
- IF $\text{error}_{(\text{temp})}$ IS negative THEN $Z_1 = y_1, Z_2 = 0$
- IF $\text{error}_{(\text{hum})}$ IS positive THEN $Z_3 = 0$
- IF $\text{error}_{(\text{hum})}$ IS near_zero THEN $Z_3 = y_0$
- IF $\text{error}_{(\text{hum})}$ IS negative THEN $Z_3 = y_1$

An adjustment is necessary for the second and fifth rules of the list. This is due to the input functions being a triangle and trapezoidal functions and two completely different inputs can present the same degree of truth. Its just necessary to contrast if variable $\text{error}_{(\text{temp})}$ is superior to 127 to determine if output Z_1 will be different to zero only when this condition is satisfied. Output Z_2 will be different from zero only when this condition is not satisfied. On the other hand only if variable $\text{error}_{(\text{hum})}$ is inferior to 127 Z_3 will be y_0 .

III. IMPLEMENTATION ON AN FPGA

In this section it is described the use VHDL to design a fuzzy controller based on the design requirements previously defined. A detailed explanation of the structure and behavior of each block is provided. Functional verification of each block was realised using software ModelSim-Altera.

A. Fuzzyfier

This module, Figure 4, has six outputs representing the degree of truth of each element regarding to negative, positive and near_zero sets, calculated for each $\text{error}_{(\text{input})}$ regarding to input functions, Figures 2 and 3. In order to model the behavior of the **Fuzzyfier** module three

options are available: CASE structure, nesting of IF structures and to calculate the property degrees using straight-line point-slope equation. To reserve the LE's for defuzzification stage a CASE structure with a 0 LE's consumption and 408,000 memory bits is used. In Figure 5 the CASE structure is shown with the corresponding input values 126, 127 and 128 in input $\text{error}_{(\text{temp})}$. The values of property were calculated taking in consideration membership functions in Figure 2 starting from the straight-line equation point-slope.

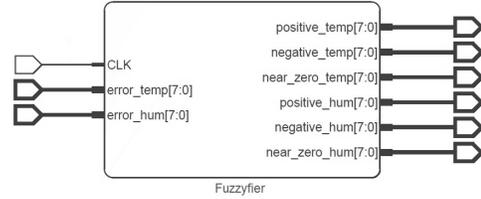


Figure 4: Structural model for the **Fuzzyfier** module.

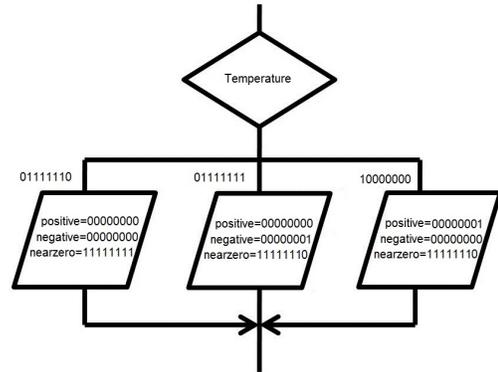


Figure 5: CASE structure for the temperature variable for values = 126, 127, 128.

B. Rule Evaluator

The structural model of **Rule Evaluator** is shown in Figure 6. Based on the inputs its calculated the consequence of each rule, applying Equation 2 and 3 and replacing the values obtained by the fuzzification module.

C. Defuzzifier

There are two widely used methods of defuzzification for Takagi-Sugeno systems, those are: Weighted Average and Mean-Max. For this paper the Mean-Max method is used, which realises an arithmetic average to the consequent of the two rules with a wider degree of implication. It is well known this method is most computationally efficient, but presents a high margin of noise in the output. In the proposed system, three rules exist to define each output Z_1, Z_2 and Z_3 . A particular characteristic for this model is that one rule always has a degree of equal implication to zero for

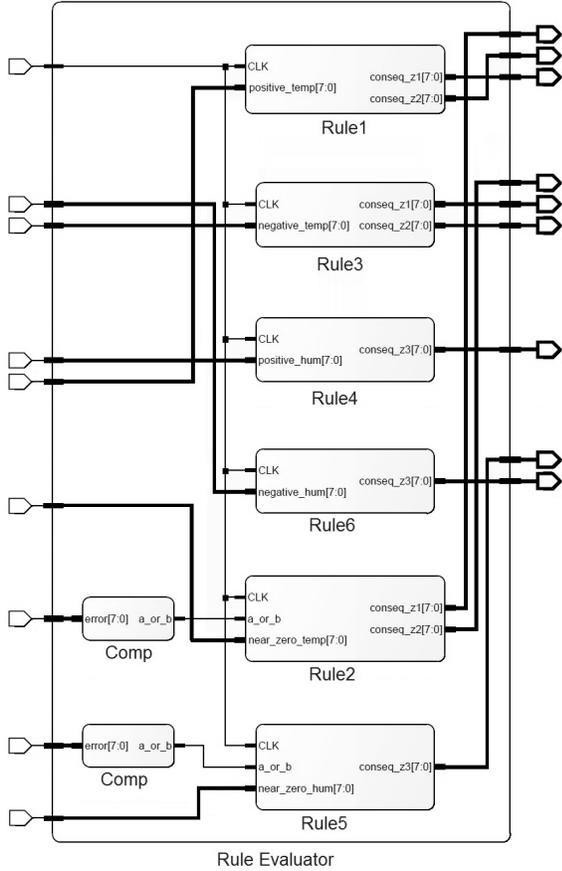


Figure 6: Structural model for the **Rule Evaluator** module.

an x input, at a moment of time t , this rule is omitted, applying the Mean-Max method with the rules that influence in the output, this phenomena can be observed in Figure 7 where the universe of discourse of the temperature variable is divided in two parts: A_1 and A_2 . As in A_1 and A_2 , just two out of three fuzzy input sets have a value different to zero. The structural model for the **Defuzzifier** module is shown in figure 8, and the structural model for the proposed controller is shown in figure 9.

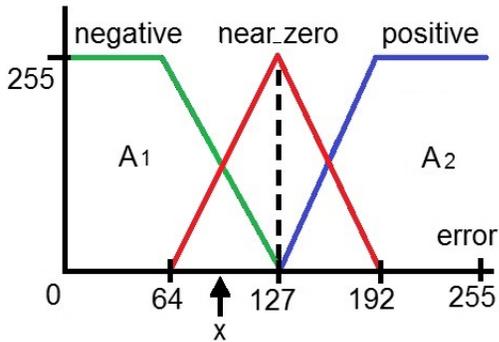


Figure 7: Input function's behavior for an x input.

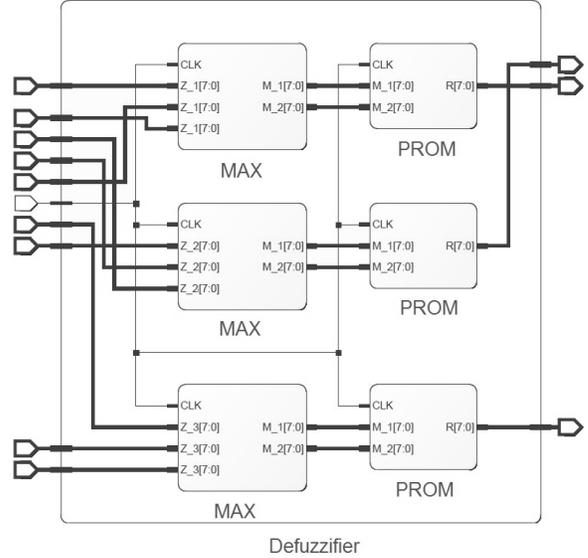


Figure 8: Structural model of the **Defuzzifier** module.

IV. SIMULATIONS

The proposed controller was implemented applying an up-down approach. All the modules were codified in VHDL and simulated using ModelSim-Alter. Quartus II Web Edition was used for the synthesis and downloads in FPGA Cyclone II EP2C35F672C6 assembled in breadboard Alters DE2. In this article the entrances for the errors of temperature and humidity were simulated using DIP switches of plate DE2 whereas the outputs: Output1, Output2 and Output3 were observed in the LED's of the plate. A test circuit was designed, to send a rank of inputs for the temperature error and humidity going from 0 to 255 to the controller, this being controlled by a 1 MHz clock. Table I shows a description of inputs and outputs of this circuit. A screen print of the results obtained during a simulation can be observed in Figure 10. Consumptions of hardware resources of this implementation are described in Table II.

To calculate the implementation's precision in FPGA, a system of similar characteristics was simulated using MATLAB & Fuzzy Logic Tool Box. Values obtained in MATLAB were considered to be the real values. Tables III, IV and V show the obtained results of this paper. First of all, system logouts (warm up, cool up and dampen) were calculated using MATLAB. In order to determine this, most representative system inputs were used (near zero inputs). Immediately after, implementation outputs in FPGA for these same input values were determined. An RMS error of 8.0290%, 7.7986% and 16.4660% for the outputs was procured. This percentage was obtained by contrasting the output in FPGA with the obtained results using Fuzzy Logic Tool Box in Matlab.

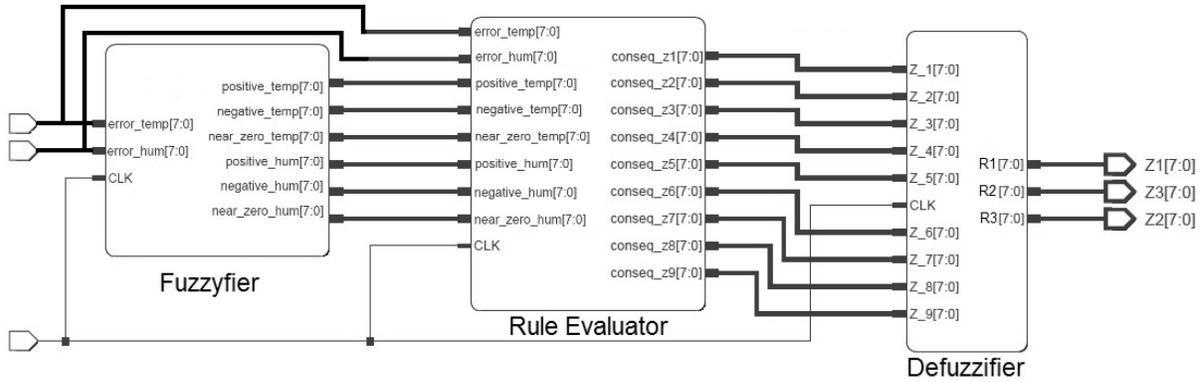


Figure 9: Structural model of the proposed controller.

Table I: DESCRIPTION OF THE USED SIGNS IN THE TEST CIRCUIT.

Number	Name	Description
1	Clk	Clock signal
2	error_temp	Temperature error (0-255)
3	error_hum	Humedad error(0-255)
4	Output1	Output (0-255 warm up)
5	Output2	Output (0-255 cool up)
6	Output3	Output (0-255 dampen)

Signal	Value	Value	Value	Value	Value
Clock					
CLK	1				
Inputs					
error(temp)	11111111	1111 111	11100110	10101001	01110110
error(hum)	01111111	0111 111	01001100	00110011	
Outputs					
Output1	00000000	0000 000			
Output2	11111111	1111 111	11100110	10101001	01110110
Output3	01111111	0111 111	01001100	00110011	

Figure 10: Controller's simulation realised in Modelsim-Altera.

Table II: HARDWARE RESOURCES CONSUME OF THE PROPOSED CONTROLLER.

LE's	Memory	Total pins
613 (1.84%)	8.19 Kbits (1.34%)	40 (25.97%)

Table III: OBTAINED RESULTS FOR THE WARM UP OUTPUT.

Input	Output1 _{MATLAB}	Output1 _{FPGA}
106	36.8471	49
111	23.7784	38
116	13.4549	27
121	5.8765	16
126	1.0431	4
128	0	0
133	0	0
138	0	0
143	0	0
148	0	0

Root Mean Square Error (RMSE) = 8.0290%

Table IV: OBTAINED RESULTS FOR THE COOL UP OUTPUT.

Input	Output1 _{MATLAB}	Output1 _{FPGA}
106	0	0
111	0	0
116	0	0
121	0	0
126	0	0
128	0.3878	2
133	4.5315	13
138	11.4311	24
143	21.0866	35
148	33.4980	47

Root Mean Square Error (RMSE) = 7.7986%

Table V: OBTAINED RESULTS FOR THE DAMPEN OUTPUT.

Input	Output1 _{MATLAB}	Output1 _{FPGA}
81	143.3667	106
86	116.5725	94
91	92.5235	83
96	71.2196	72
101	52.6608	61
106	36.8471	49
111	23.7784	38
116	13.4549	27
121	5.8765	16
126	1.0431	4

Root Mean Square Error (RMSE) = 16.4660%

The developed FPGA implementation works with integer numbers and uses the Mean-Max method in the defuzzifier unit. This is due to the reduced resources consumption that it implies, although as it can be observed this adjustment represents an increase of the error in the output. In the selected application these errors are imperceptible, but if these type of errors are not permitted in the used system, the controller can easily be adapted to work with the average sum method in the defuzzifier module, in Table VI, it can be observed how the error has been reduced considerably, although the consumption of resources has been increased, see Table VII.

Table VI: OBTAINED RESULTS FOR THE WARM UP OUTPUT APPLYING THE WEIGHTED AVERAGE METHOD IN THE DEFUZZIFIER MODULE.

Input	Output _{MATLAB}	Output _{FPGA}
81	143.3667	144
86	116.5725	117
91	92.5235	93
96	71.2196	72
101	52.6608	53
106	36.8471	37
111	23.7784	24
116	13.4549	14
121	5.8765	6
126	1.0431	2
127	0.00	0
128	1.0431	2
129	5.8765	6
130	13.4549	14
131	23.7784	24
132	36.8471	37

Root Mean Square Error (RMSE) = 0.5325%

Table VII: HARDWARE RESOURCES CONSUME OF THE PROPOSED CONTROLLER USING WEIGHTED AVERAGE METHOD IN THE DEFUZZIFIER MODULE.

LE's	Memory	Total pins
1622 (4.88%)	8.19 Kbits (1.34%)	40 (25.97%)

V. CONCLUSIONS

The proposed controller processes variables of temperature and humidity in parallel, this improve considerably the response time regarding to a controller of similar characteristics implemented on a PIC, DSP or MATLAB; making this FPGA implementation a viable option to apply on systems in need of realise a high speed processing.

The application selected in this paper was the control of a chicken incubator. Due to the flexibility presented by HDL it is possible to redesign the controller from new design parameters without incurring in nonrecurring engineering and without applying any changes in the internal structure of the controller.

During this research variables of temperature and humidity were controlled, nevertheless, considering the parallelism presented by HDL, is possible escalate the proposed methodology for n variables.

REFERENCES

- [1] F. Yasin, A. Tio, M. Islam, M. Reaz, and M. Sulaiman, "The hardware design of temperature controller based on fuzzy logic for industrial application employing fpga," *Microelectronics. The 16th International Conference on*, pp. 157–160, 12 2004.
- [2] D. Kim, "An implementation of fuzzy logic controller on the reconfigurable fpga system," *Industrial Electronics, IEEE Transactions on*, vol. 47, pp. 703–715, 06 2000.
- [3] Z. Runjing, X. Hongwei, and W. Zhijun, "Design of fuzzy temperature control system based on fpga," *Control, Automation and Systems Engineering (CASE), International Conference on*, pp. 1–4, 07 2011.
- [4] M. Poplawski and M. Bialko, "Implementation of parallel fuzzy logic controller in fpga circuit for guiding electric wheelchair," *Human System Interactions, Conference on*, pp. 405–408, 05 2008.
- [5] C.-F. Juang and J.-S. Chen, "Water bath temperature control by a recurrent fuzzy controller and its fpga implementation," *Industrial Electronics, IEEE Transactions on*, vol. 53, pp. 941–949, 06 2006.
- [6] E. Parrilla, J.-R. Torregrosa, J. Riera, and J.-L. Hueso, "Fuzzy control for obstacle detection in stereo video sequences," *Mathematical and Computer Modelling: An International Journal*, vol. 54, pp. 1813–1817, 05 2011.
- [7] D. Oliveira, G. de Lima Henn, and O. da Mota Almeida, "Design and implementation of a mamdani fuzzy inference system on an fpga using vhdl," *Fuzzy Information Processing Society (NAFIPS), Annual Meeting of the North American*, pp. 1–6, 07 2010.
- [8] S. Uppalapati and D. Kaur, "Design and implementation of a mamdani fuzzy inference system on an fpga," *Fuzzy Information Processing Society (NAFIPS), Annual Meeting of the North American*, vol. 2010, pp. 1–6, 06.
- [9] O. Montiel, Y. Maldonado, R. Sepúlveda, and O. Castillo, "Simple tuned fuzzy controller embedded into an fpga," *Fuzzy Information Processing Society (NAFIPS), Annual Meeting of the North American*, pp. 1–6, 05 2008.
- [10] M. McKenna and B. Wilamowski, "Implementing a fuzzy system on a field programmable gate array," *Neural Networks. International Joint Conference on*, vol. 1, pp. 189–194, 07 2001.
- [11] T. Sutikno, A. Z. Jidin, A. Jidin, and N. R. N. Idris, "Simplified vhdl coding of modified non-restoring square root calculator," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 1, pp. 37–42, 03 2012.
- [12] N. Zhang, R. Kamdem, E. Ososanya, W. Mahmoud, and W. Liu, "Vhdl implementation of the hybrid fuzzy logic controllers with fpga," *Intelligent Control and Information Processing (ICICIP), International Conference on*, pp. 13–15, 08 2010.