

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/257139762>

FPGA Fuzzy Controller Design for Magnetic Ball Levitation

Article · October 2012

DOI: 10.5815/ijjsa.2012.10.08

CITATIONS

5

READS

251

2 authors, including:



Basil Hamed

Islamic University of Gaza

27 PUBLICATIONS 175 CITATIONS

SEE PROFILE

FPGA Fuzzy Controller Design for Magnetic Ball Levitation

Hosam Abu Elreesh

Electrical Engineering Department, Islamic University of Gaza, Gaza, Palestine
Email: m_hossam@hotmail.com

Basil Hamed

Electrical Engineering Department, Islamic University of Gaza, Gaza, Palestine
Email: bhamed@iugaza.edu

Abstract— this paper presents a fuzzy controller design for nonlinear system using FPGA. A magnetic levitation system is considered as a case study and the fuzzy controller is designed to keep a magnetic object suspended in the air counteracting the weight of the object. Fuzzy controller will be implemented using FPGA chip. The design will use a high-level programming language HDL for implementing the fuzzy logic controller using the Xfuzzy tools to implement the fuzzy logic controller into HDL code. This paper, advocates a novel approach to implement the fuzzy logic controller for magnetic ball levitation system by using FPGA.

Index Terms— Fuzzy Control, PI, FPGA, Magnetic Levitation Ball, VHDL

I. Introduction

In the recent years Fuzzy controller is used to control complex engineering problems which are difficult to solve by classical methods. Finding many different hardware implementations of fuzzy logic systems (FLSs), general-purpose microprocessors and microcontrollers are mostly used for implementing FLS in hardware, but with the complex systems these devices cannot perform operations assigned to it as required. In recent years many studies emerged illustrate the different ways to implement fuzzy control using FPGA in different application. The advantage of using FPGA is suitable for fast implementation and quick hardware verification. FPGA based systems are flexible and can be reprogrammed unlimited number of times. J.E. Bonilla, V.H. Grisales and M.A. Melgarejo [1]; the fuzzy controller architecture in this paper focused on the treatment of errors and changes in errors with tuning gains. This paper presented the development of an FPGA-based proportional-differential (PD) fuzzy LUT controller. The fuzzy

inference used a 256-value LUT. This method was used due to its reduced computation time cost. McKenna and Wilamowski [2] have investigated method to implement fuzzy logic controller (FLC) on a field FPGA and obtained very smooth control surfaces. Vuong et al [3]; described a methodology of implementing FLS using very high speed integrated circuit hardware description language (VHDL). The main advantages of using HDL are rapid prototyping and allowing usage of powerful synthesis tools such as Xilinx ISE, Synopsys, Mentor Graphic, or Cadence to be targeted easily and efficiently. Patyra and Grantner [4]; presented a paper investigating design issues for digital fuzzy logic system (FLS) circuits. In their study, comparisons between the current trends were conducted and they proposed a new methodology whereby a fully parallel architecture is employed to achieve high performance in hardware implementation of digital FLSs. They presented ways to translate an FLS into hardware, and discussed methods for testing the FLS hardware performance. Both SISO and MIMO FLC hardware implementations were presented. Their proposed methodology provides an improved solution for high-speed, real-time applications.

II. Fuzzy Control

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth. There are not two values (true or false) but there are two limits (1) completely true and (0) completely false and the result can have different degree between these limits [5]. Fuzzy Control applies fuzzy logic to the control of processes by utilizing different categories, usually 'error' and 'change of error', for the process state and applying rules to decide a level of output. There are many models of FLC, but the most famous are the Mamdani model, Takagi-Sugeno-Kang (TSK) model and Kosko's additive model (SAM) [5]. This paper uses Mamdani model.

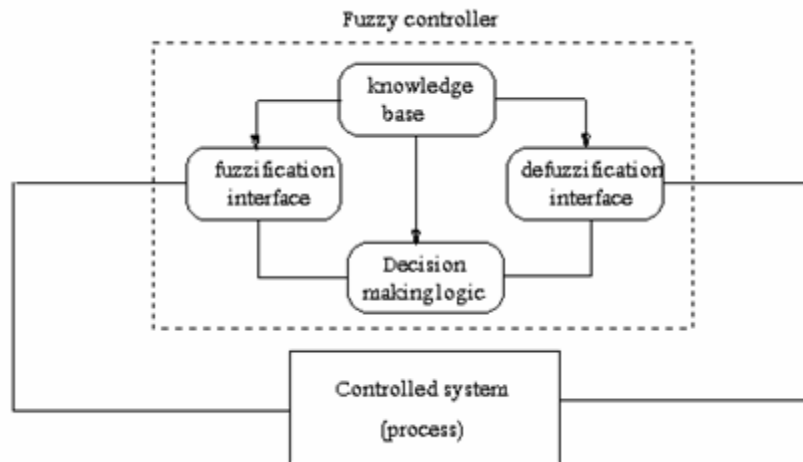


Fig. 1: Mamdani Model

Fig.1 shows the block diagram of Mamdani fuzzy system model, the rule base of this model is in the next form.

IF (X is A) and (Y is B) ... THEN (Z is C).

Where A, and B are membership of the inputs, C is membership of the output as shown in Fig.1. Mamdani model block consist of three stages.

- **Fuzzification:** - Fuzzification means converting a crisp value of process variable into a fuzzy set. In order to make it compatible with the fuzzy set representation of the process state variable.
- **Fuzzy Associative Memory (FAM):** - FAM is a set of fuzzy associations between the input and the output [6]. This stage consists of two parts:
 - A. **Knowledge base:** - Knowledge base contains a data base and rule base. Data base provides necessary definitions for linguistic rules, and the rules base consist of the IF-THEN rules, which can be derived by using four ways:
 1. Expert Experience and control engineering knowledge.
 2. Based on fuzzy modeling of human operators central action.
 3. Based on learning
 4. Based on fuzzy model of a process.
 - B. **Decision-Making:** - Decision-Making means choosing the most appropriate action from several possible actions.
 - **Defuzzification:** - Defuzzification strategy is aimed at producing a non-fuzzy control action, or we can say defuzzification means the conversion of the fuzzy output values into crisp values.

III. FPGA

FPGA is digital integrated circuits (ICs) that have electronics blocks which can be programmed, and these blocks have configurable interconnection between them. These blocks can be used by the designed engineer to perform a huge range of tasks. There are two kinds of FPGA; one kind can be programmed for single time only and it so called one time programmable (OTP). The second kind can be reprogrammed many times. Since 1960, the ICs are used in the human life, and there are many different types of ICs that have been used such as memory devices, microprocessors (μP), Programmable logic devices (PLDs), application-specific integrated circuits (ASICs), application-specific standard parts (ASSPs), and FPGAs. Fig.2 shows the timeline of the ICs and other devices.

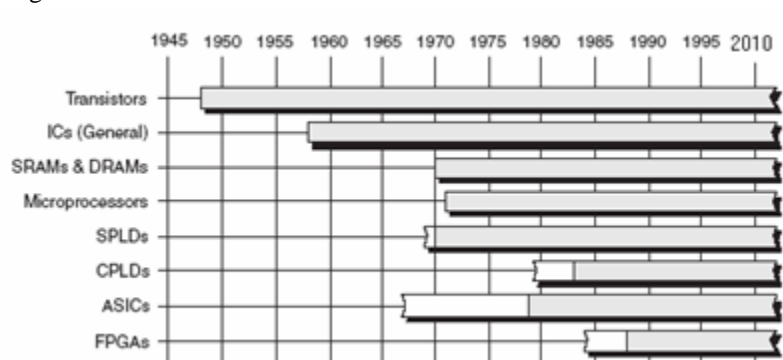


Fig. 2: Technology Timeline

The white parts in the timeline means the time that the devices are found but were not used by the engineering widely. For example Xilinx has been designed first FPGA in 1984 but started being in use by the engineers in 1990 [7]. Every FPGA has three major components, configurable logic blocks (CLBs), input/output blocks (IOBs), and interconnects, figure 3 shows these blocks. CLBs are responsible for building the logical circuit for the user. IOBs are responsible for the interface between package pins and internal signal lines. Interconnects are responsible for routing paths to connect the inputs and outputs of the CLB and IOB [8].

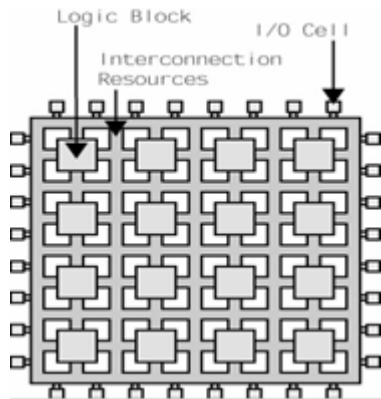


Fig. 3: FPGA Architecture

Manufacturers use different technologies for the implementation in FPGA. Examples of these technologies are:

- 1- EPROM technology:
- 2- EEPROM technology.
- 3- Technology Static Ram (SRAM).
- 4- The FLASH technology.

These days, FPGAs offer the possibility of using dedicated blocks such as memories RAM, multipliers cabled PCI interfaces and processor cores. The design of control architectures is done using CAD tools. There are two commonly used languages, Very high speed integrated VHDL and Verilog. These two languages are standardized and compatible with all FPGA technologies previously introduced. Fig.4 shows the FPGA programming steps. This thesis will use the Spartan 3e FPGA from Xilinx company.

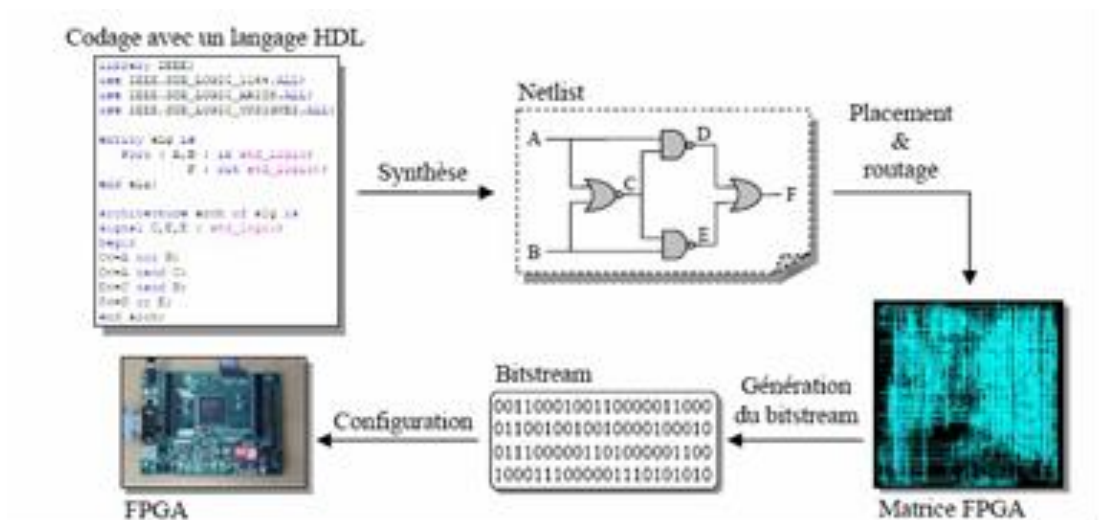


Fig. 4: Programming a FPGA

IV. Fpga And Fuzzy Implementation

There are many researches of FLC implementation on FPGA [2][3][4]. These researches discussed the implementation of FLS onto FPGA, but writing VHDL code is not easy. Because fuzzy systems have many correlated parameters, writing VHDL code needs good knowledge in VHDL language. Reducing design time of FLS causes focusing on optimizing the FLS to give high performance. Software CAD tools are used to deal with the complex tasks of FLS. Some of these tools are [9]:

- 1-FuzzyTech.

2-Rigel.s Fuzzy Logic Applications Software Helper (rFLASH).

3-Fuzzy Inference Development Environment (FIDE).

These tools cannot generate a VHDL code that work with FPGA devices. But since 1992 XFUZZY CAD tool has been developed, and has evolved with the passage of time. This paper uses Xfuzzy to generate VHDL code for FLC.

V. Xfuzzy Cad Tools

Xfuzzy is a CAD tool that was developed using JAVA language. Xfuzzy tool can be used with two different operating systems (Windows and UNIX). It

offers the safety advantages of Java programs. This tool is a combination of several tools that covering the four stages of FLS design: description, tuning, verification, and synthesis stages. Fig.5 shows the Interface Applied of the Xfuzzy [9][10][11][12].

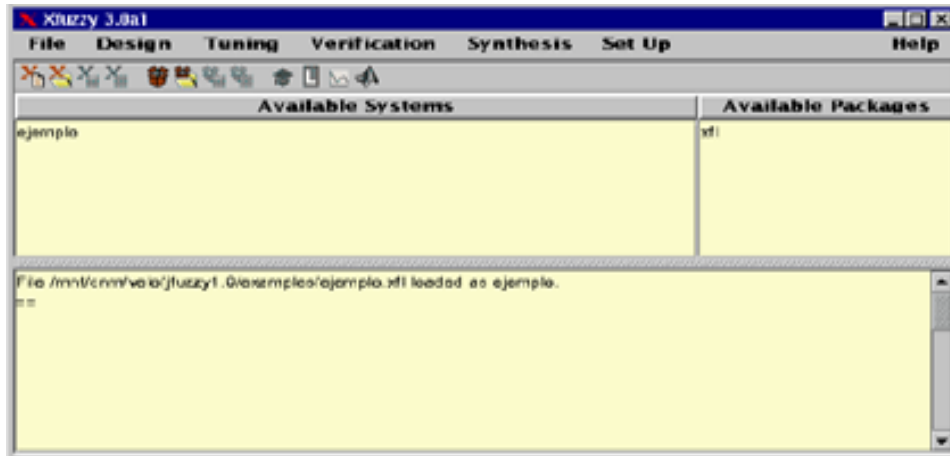


Fig. 5: Main Window of Xfuzzy 3.0.

1. Description stage

Xfuzzy has two tools that are used in this stage (*xfedit* and *xfpkg.*). The first tool dedicated to the definition of linguistic variables and the logical relations between them. The second tool *xfpkg* responsible for all mathematical function of the *xfedi*.

2. Tuning Stage

This stage uses *xfsl* tool. To work with this tool, users must have two types of data, training data and testing data. This tool works only if the designer will use fuzzy systems as classifier.

3. Verification Stage

This stage work as simulation stage which be used to study the behavior of the system, and detect the potential deviations from behavior and the source of this deviations. Xfuzzy has four verification tools for these purposes: *xf2dplot*, *xf3dplot*, *xfm*, and *xfsim*.

4. Synthesis Stage

It is the final stage of the FLS design, in this stage the FLS are converted to different codes depending on the software or hardware target that the designer will use fuzzy system with. This tool can convert the FLS to different codes, these codes is C code, Cpp code, java code, sysgen code that work with Xilinx toolbox in Matlab Simulink, and finally VHDL code that used in FPGA devices .

VI. Simulation

The CE 152 Magnetic Levitation Model is one of the ranges of educational scale models offered by Humusoft Company for teaching system dynamics and control engineering principles. The model belongs to the range of teaching systems directly controllable by a PC computer in real time. The CE 152 Magnetic Levitation model is one dimensional strongly unstable system designed for studying system dynamics and experimenting with number of different control algorithms based on classical and control theory. Fig.6 shows CE 152 model [13].

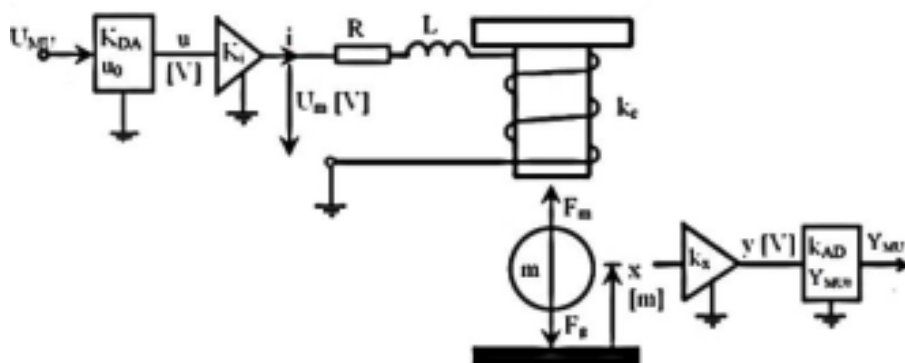


Fig. 6: CE152 Magnetic Levitation Model

The model shown in Fig.6 consists of the following blocks:

1. D/A converter.
2. Power amplifier.
3. Ball & coil subsystem.
4. Position sensor.
5. A/D converter.

The mathematical equation of this model is:

$$\begin{bmatrix} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{i^2 k_c}{m_k (x_1 - x_0)^2} - g - \frac{k_f x_2}{m_k} \\ \dot{x}_3 = I \end{bmatrix} \quad (\text{nonlinear state space model})$$

Where

x_1 is the ball position, x_2 ball velocity and x_3 coil current

m_k = ball mass [kg].

g = gravity acceleration constant [$m.s^{-2}$].

k_f = viscose friction.

x_0 = coil bias [m].

K_c = coil constant.

6.1 Fuzzy Controller Design for CE152 Model

To apply the fuzzy logic controller to the magnetic levitation CE 152, certain properties of the system are exploited so that the design of the controller can be made easier. As the system is symmetrical, it is assumed that symmetrical membership functions about the y-axis will provide a valid controller. A symmetrical rule-base is also assumed. The fuzzy controller of magnetic levitation uses Mamdani model. The FLC has two inputs which are error and change of error and the output is the change of voltage. Figure 7 shows the membership functions of fuzzy controller using Fuzzy

Toolbox of Matlab software. The ranges of the inputs and output is [-1 1]. All have 7 membership functions.

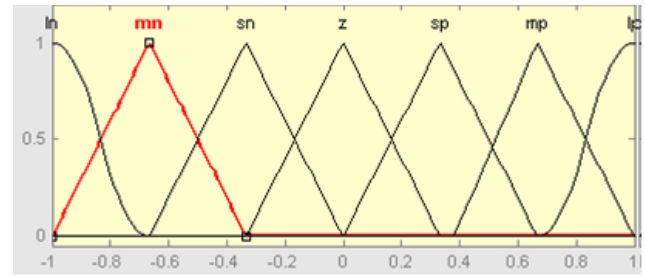


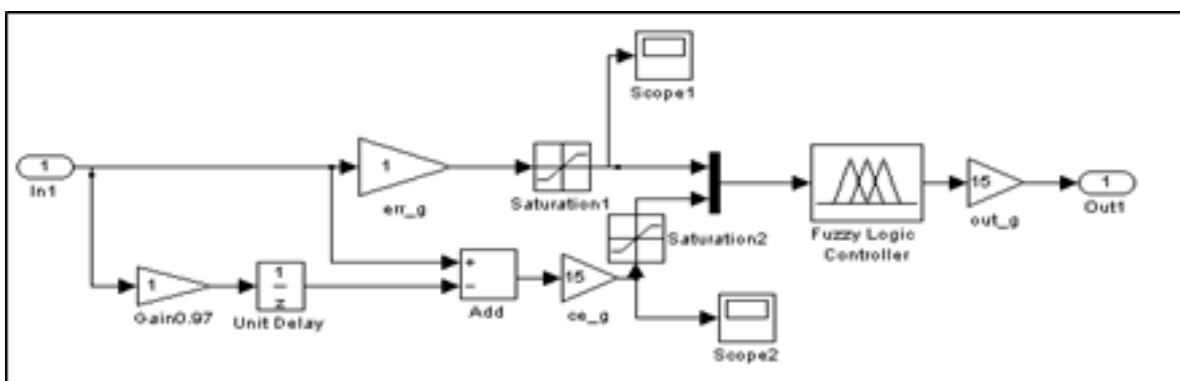
Fig. 7: Memberships of Inputs and Output Fuzzy Controller

The next step after designing the membership functions is to write the rules of the fuzzy controller. These rules are chosen based on knowledge base and experts. Table 1 shows the rules base as a matrix

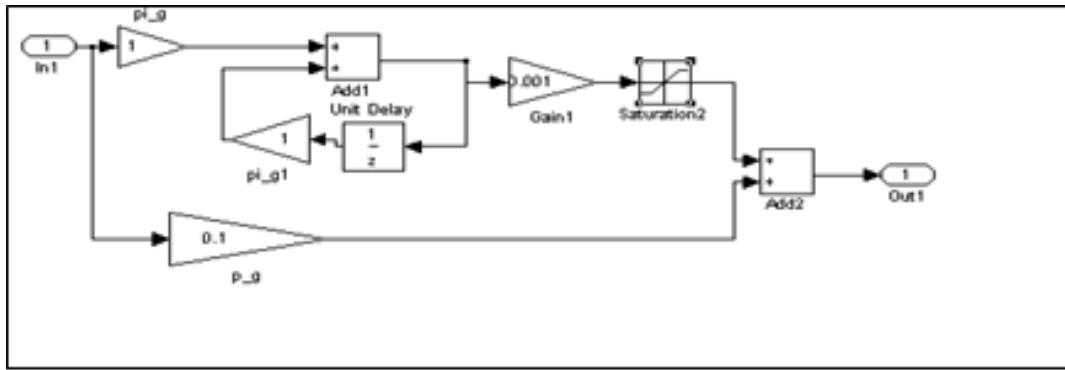
Table 1 Fuzzy Control Rule Base of Magnetic Levitation

e	ln	mn	sn	z	sp	mp	lp
ce	ln	mn	sn	z	sp	mp	lp
ln	ln	ln	ln	ln	mn	sn	z
mn	ln	ln	ln	mn	sn	z	sp
sn	ln	ln	mn	sn	z	sp	mp
z	ln	mn	sn	z	sp	mp	lp
sp	mn	sn	z	sp	mp	lp	lp
mp	sn	z	sp	mp	lp	lp	lp
lp	z	sp	mp	lp	lp	lp	lp

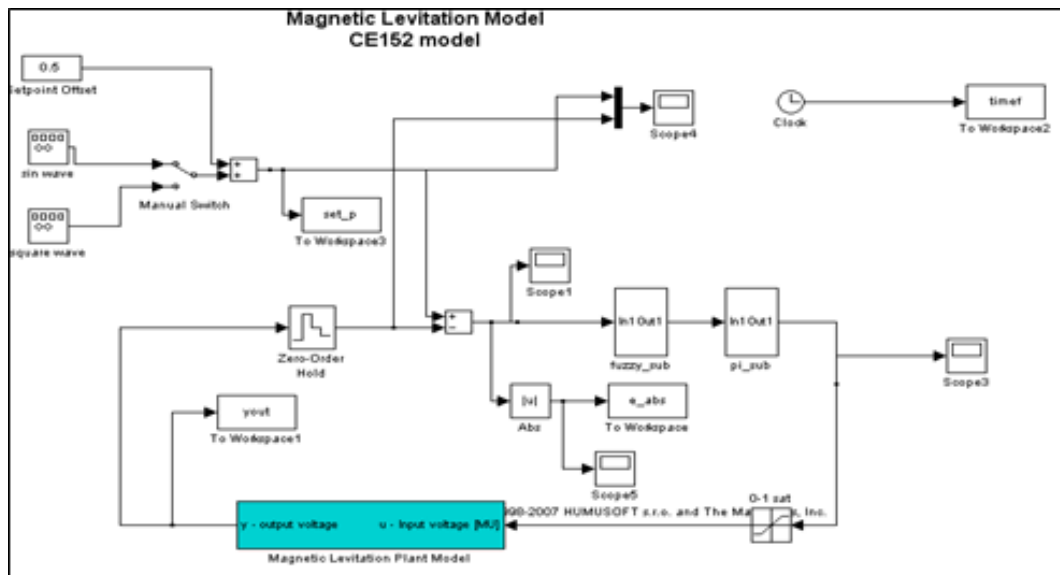
After designing fuzzy controller, the controller will be connected to the CE152 magnetic levitation Matlab Simulink model. Figure 8 a, b and c shows the fuzzy controller, proportional integral controller (PI) subsystems and all system respectively. The integral part is used to convert the change of voltage value that come from fuzzy controller to real voltage which will act on the magnetic levitation CE152. Using tuning method the good proportional gain is 0.1 and the integral gain is 1.



(a)



(b)



(c)

Fig. 8: a) Fuzzy Controller Subsystem - b) PI Subsystem –c) Magnetic Levitation Model

Fig.9 shows the output of the magnetic levitation after connecting to the fuzzy controller. The set point is unit step has value 0.5 which is in the center of the gap; this point is one of the equilibrium points of magnetic levitation CE152 model. There is no overshoot and the settling time is nearly 0.2 sec, rising time is 0.106 sec.

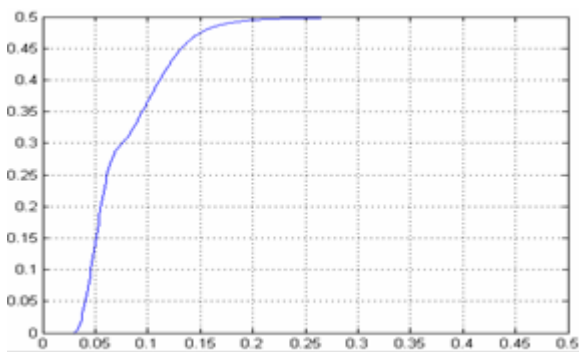


Fig. 9: Step Response of the System

Fig.10 and Fig.11 show the output response with two additional set points. These figures show that the fuzzy controller can keep the stability of the system with various set points.

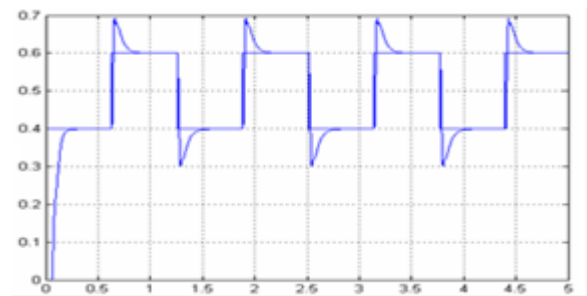


Fig. 10: Square Wave Output Response of the System

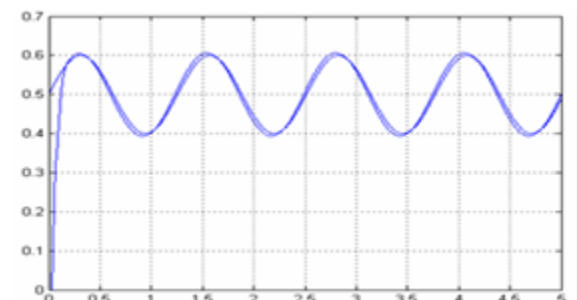


Fig. 11: Sine Wave Output Response of the System

6.2 VHDL Fuzzy Controller Implementation

Fig.12 shows the block diagram of the controller using FPGA. It consists of three main parts, ADC, FPGA, and DAC. ADC and DAC that will be used in

this thesis have 8 bits resolution. The integrated circuits (ICs) that will be used are PIC 16f877 as ADC, and DAC 0800. These two ICs have eight bits parallel interface with FPGA.

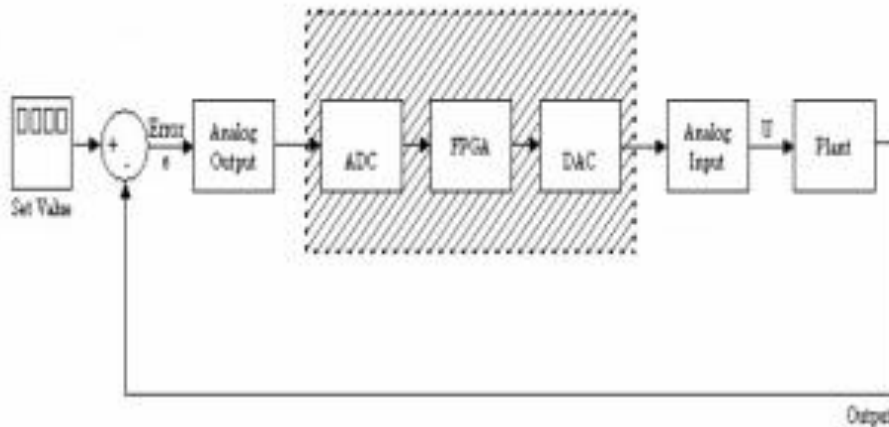


Fig. 12: Block Diagram of FPGA and its Interface with the System

The fuzzy controller consists of three parts:

- 1- Summation part has two inputs set point and feedback, and generates the error and change of error that are inputs to fuzzy controller part.
- 2- Fuzzy controller is generated using Xfuzzy CAD tool and has two inputs (error and change of error) and one output (change of voltage).

- 3- PI (proportional-integral) part has one inputs (change of voltage) and one output (effective voltage).

Fig.13 shows the fuzzy controller membership for error input using Xfuzzy tool (the two inputs error and change of error have same membership)

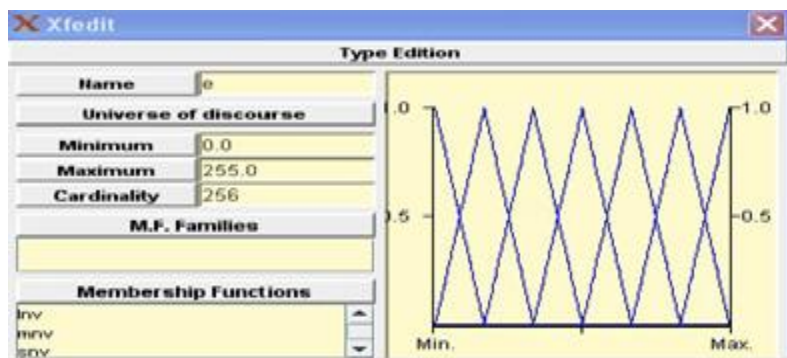


Fig. 13: Error Input Using Xfuzzy Tool

Fig.14 and 15 show the membership function for the output of the fuzzy controller (cv), rule base respectively.

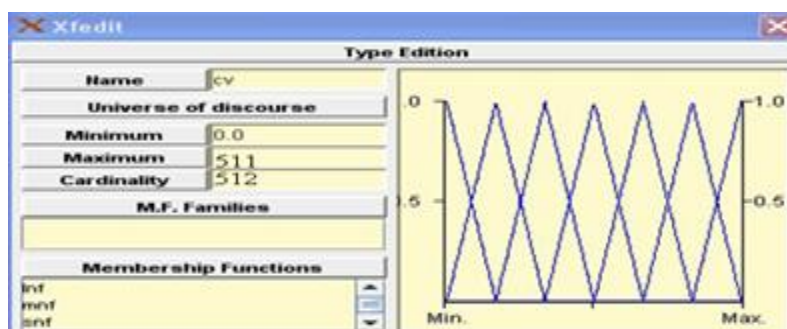


Fig. 14: Change of Voltage Output Using Xfuzzy Tool

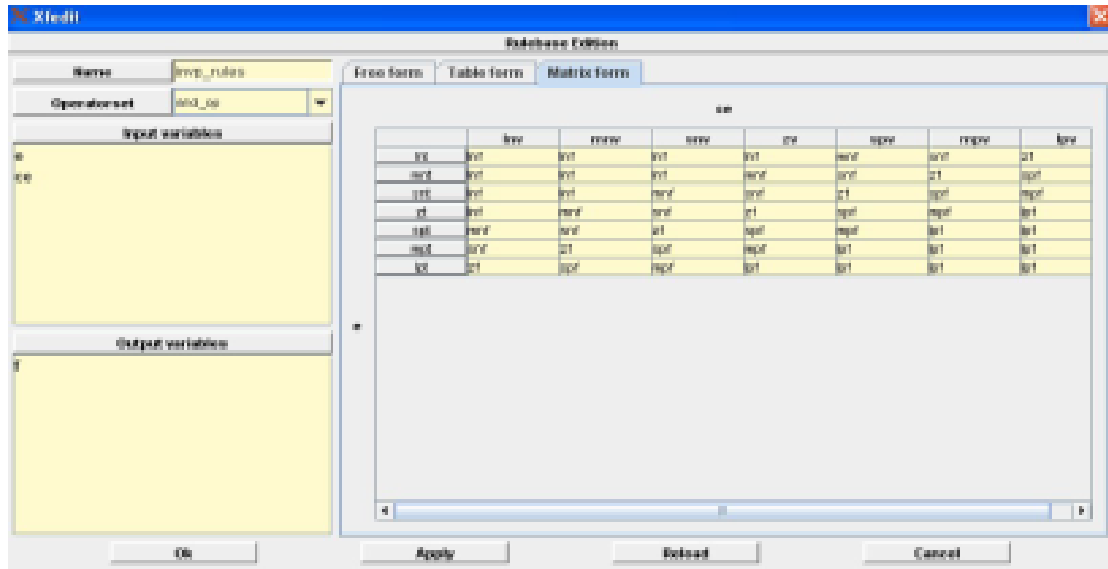


Fig. 15: Fuzzy Rules Xfuzzy Tool

The last step after design the fuzzy controller using Xfuzzy tool is generating the vhdl code for this controller; this code is used with "sum.vhd" and "pi.vhd" files to complete the controller. These files will be used with ISE 10.1 software to generate bit file that is uploaded to FPGA. Xfuzzy tool will generate System

Generator blocks of fuzzy controller that is used with Matlab simulink. "Black Box" block that is in the System Generator Matlab toolbox will be used to insert the "sum.vhd" and "pi.vhd" into Matlab Simulink as shown in Figure 16 and 17 respectively

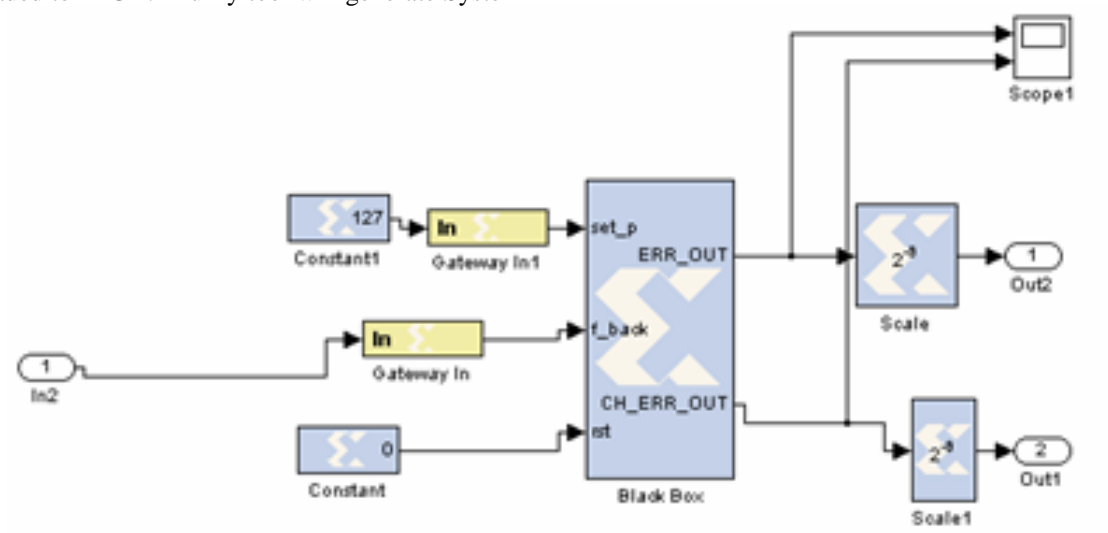


Fig. 16: Block Diagram of Summation Subsystem

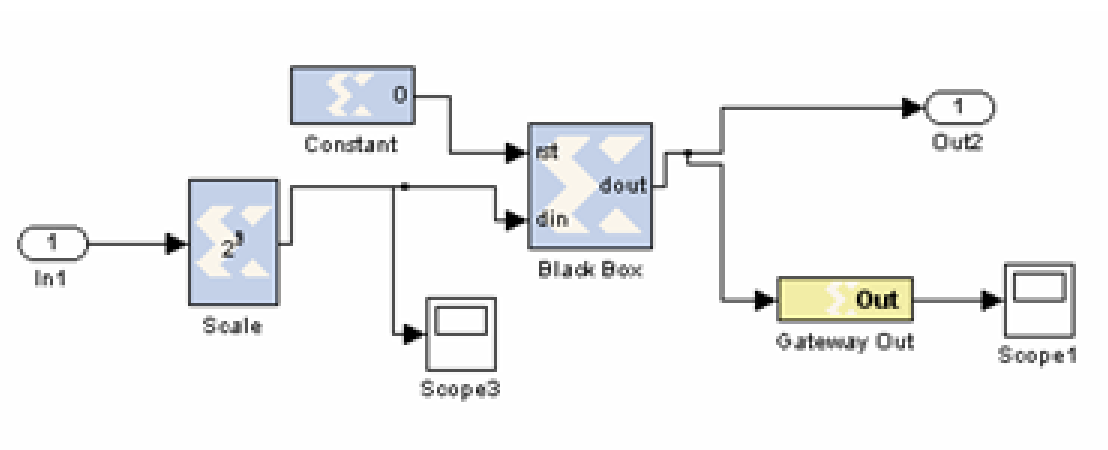


Fig. 17: Block Diagram of PI Subsystem

Fig.18 shows the blocks of fuzzy controller that is generated using Xfuzzy tool (sysgen code).

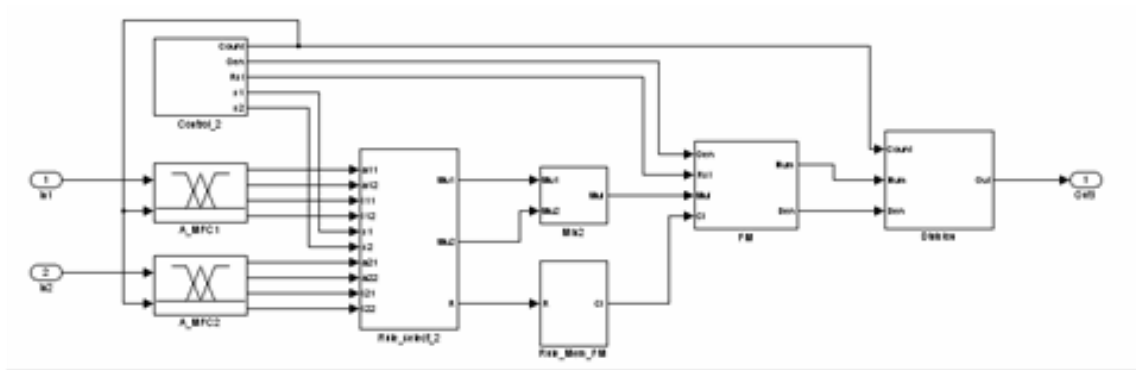


Fig. 18: Block Diagram of Fuzzy Controller Subsystem

Fig.19 shows the final system after connecting all sub systems

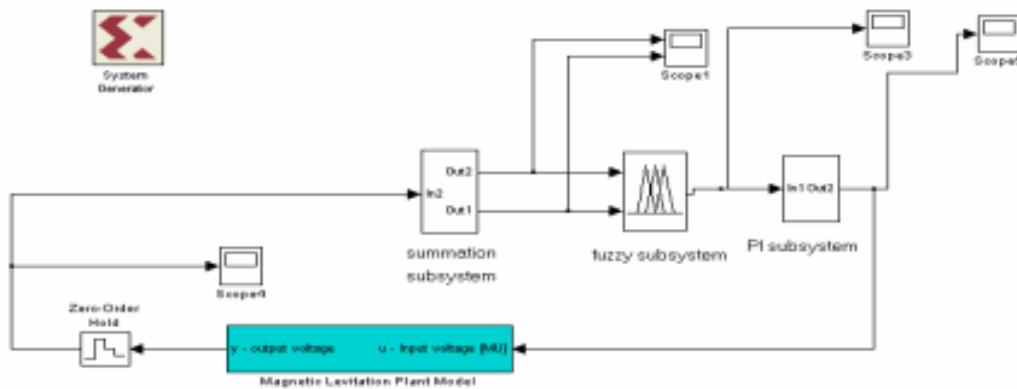


Fig. 19: Block Diagram of Fuzzy Controller Using VHDL Code

After building the Simulink model of fuzzy controller using System Generator toolbox, the system runs at three different set points, step, sine wave and square wave to test the VHDL code if it works properly or not. Fig.20 shows the output response of the system at set point equal 127. (0.5 at normalize mode). Fig.21 shows the error and change of error of the ball position, the two values are shifted by 255, because the VHDL code of fuzzy controller works only in the interval [0 510], Where 0 means that the -255 and 510 means 255, so 0 error in Matlab equal 255 in VHDL code. We need to set the simulation time in System Generator block to be 0.00000002 sec to work as real FPGA because the FPGA clock that is used in VHDL code is 50 MHz

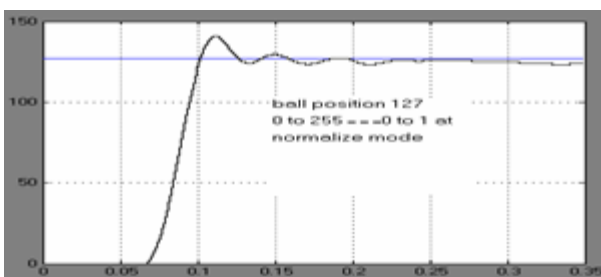


Fig. 20: Step Response of the System Using VHDL Code

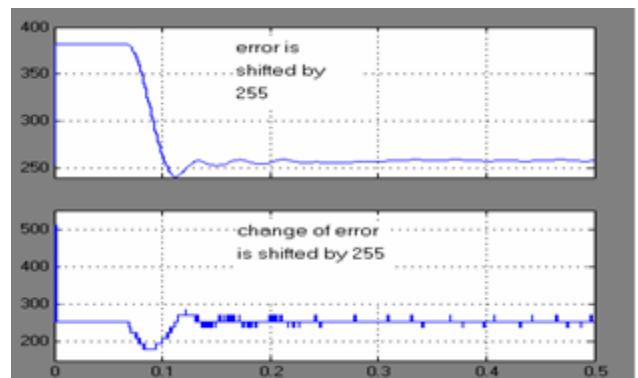


Fig. 21: Error and Change of Error Using VHDL Code

Fig. 22 and Fig.23 show the output response when the set point is sine wave and square wave respectively.

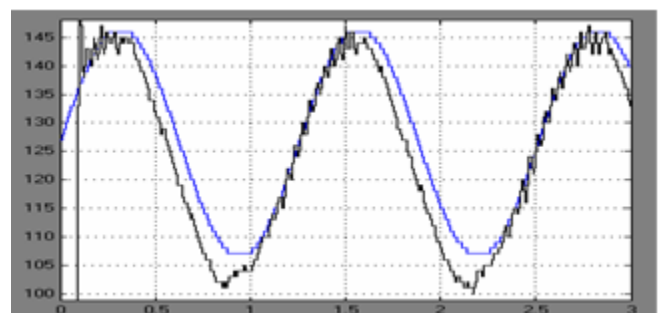


Fig. 22: Sine Wave Output Response Using VHDL Code

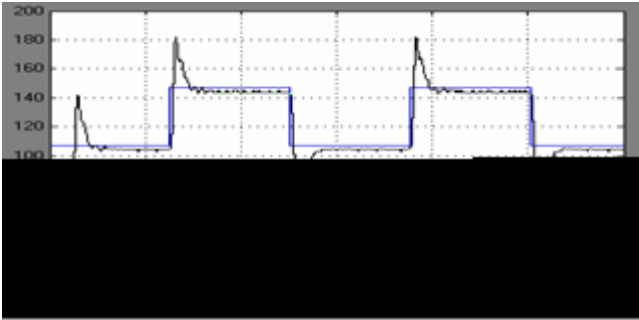


Fig. 23: Square Wave Output Response Using VHDL Code

VII. Conclusion

The real time implementation of the fuzzy logic controller for the various driving conditions and terrains has been achieved on a Xilinx Spartan 3E FPGA using VHDL. In this paper the magnetic levitation CE152 Model is used as practical example of nonlinear systems. The fuzzy controller was designed with Matlab software and this controller was tested with the CE152 Model. The fuzzy controller stabilized the magnetic levitation CE152 model under different set points. The Xfuzzy tool was used to generate the VHDL code of fuzzy controller, this code was used with "sum.vhd" and "pi.vhd" VHDL codes to give the overall controller of Magnetic Levitation CE152 Model. These codes were tested using Matlab/Simulink under different set points and the results were good. After design the VHDL codes of fuzzy controller this code was programmed and tested using Spartan 3E FPGA and the CE152 Model was stable

References

- [1] J.E. Bonilla, V.H. Grisales and M.A. Melgarejo, "Genetic tuned FPGA based PD fuzzy LUT controller," IEEE International Conference. Fuzzy Systems, Vol 3, pp :1084 – 1087, 2001.
- [2] Michael McKenna and Bogdan M. Wilamowski, "Implementing a Fuzzy System on a Field Programmable Gate Array," IEEE International Joint Conference on Neural Networks, Volume: 1, p.189-194, 2001.
- [3] T. Philip, M. Vuong, asad, Madni and b. jim Vuong, . VHDL implementation for a fuzzy logic controller., BEI technologies, inc. 13100 telfair avenue, sylmar, 2006.
- [4] MJ. Patyra & J.L. Grantner, "Hardware Implementation of Digital Fuzzy Logic Controller,". Information Science. An International Journal (113): pp.19-54, 1999.
- [5] John Yen & Langari Reza, "Fuzzy Logic Intelligence Control and Information," Prentice-Hall, Englewood Cliffs, 1999.
- [6] Prasad, Ram., "Fuzzy Logic Control," class handout, New Mexico State University, Electrical & computer Engineering, 1996.
- [7] Maxfield, Clive, "The Design Warrior's Guide to FPGAs," Newnes, 2004.
- [8] Balasaheb S. Darade, Abhishek Singh Chauhan and Tarun A.Parmar, "Paper Presentation On Programming FPGA's Using Handel-C," Jawaharlal Nehru Engineering College Aurangabad.
- [9] K. Mabasa, Mohamed Akil, Thierry Grandpierre, B.J. Van Wyk and M.A. Van Wyk "Automatic VHDL Code Generation for Fuzzy Logic Systems," African Journal of Science and Technology, 2008.
- [10] Gersnoviez and M. Brox, "Using Xfuzzy Environment for the Whole Design of Fuzzy Systems," Proc. IEEE International Conference on Fuzzy Systems, London, July 23-26.2007 .,
- [11] D. R. Lopez, S. Sanchez-Solano, and A. Barriga, "Xfuzzy: A Design Environment for Fuzzy Systems," Proc.7th IEEE International Conference on Fuzzy Systems, pp.1060-1065, Anchorage, May 1998..
- [12] F.J.M. Velo, L. Baturone, S.S. Solano and A. Barriga, "Rapid design of fuzzy systems with Xfuzzy," Proceedings of the IEEE 12th International Conference on Fuzzy Systems, May 25-28, Sevilla, Spain, pp: 342-347.2003.
- [13] CE 152 magnetic levitation model-education manual. Humusofts.r.o 2002.



Hosam Abu Elreesh received B.S degree in Electrical Engineering and M.S degree in Control Systems both are from Islamic University of Gaza, Palestine in 2000 and 2011 respectively. His current researches are Control Systems, Fuzzy Logic Controllers, FPGA, and Genetic Algorithms.



Dr. Basil Hamed is Associate Professor of Electrical Engineering Department, Islamic University of Gaza, Palestine, since 1999. He has Bachelor Degree in Electrical Engineering from New Mexico State University, NM. USA in the year of 1989, he received Master degree from University of New Orleans, La. USA in the year of 1992, and earned his PhD (Fuzzy Control System) from New Mexico State University, NM USA in the year 1999. He has 15 years of teaching experience and has published many papers in national and international journals. His fields of interest include Control Systems, Fuzzy Control, Simulation & Modeling, FPGA, Genetic Algorithm, SCADA System, Robotics, Signal and Image Processing.