# Analysis and Performance Evaluation of PD-like Fuzzy Logic Controller Design Based on Matlab and FPGA

Zeyad Assi Obaid, *Member, IAENG,* Nasri Sulaiman, M. H. Marhaban and M. N. Hamidon, *Member, IAENG*

*Abstract*- **This paper presents an analysis and performance evaluation of the proportional-derivative (PD) fuzzy logic controller design by using Matlab and field programmable gate array (FPGA). The fuzzy logic controller consists of a Fuzzifier, inference engine and Defuzzifier; the Fuzzifier block accepts two PD inputs. Two types of controller are designed; the first one is using fuzzy logic toolbox in Matlab. The second type is designed using *VHDL* language for implementation on FPGA. Mathematical models of robot arm and bench-top helicopter are used for the purpose of simulation with the first type. This controller is used with a unity feedback control system in Matlab Simulink, in order to control these systems and to generate the simulation results. The best response with the robot arm has 0.02 errors and zero overshot, and the best response with the bench-top helicopter has 0.01 error with 0.001 overshot. Altera Quartus II and ModelSim simulation program are used to generate the simulation results of the second type. A mathematical model that represents industrial processes, such as temperature, pressure, pH, and fluid-level controls with unity feedback control systems and subjected to 0.2 step input is used to generate these results. This FPGA-based controller is able to produce a fast response ranging from 0.3 μs, even with time delay added with the plant model.**

*Index Terms*- **Fuzzy Logic controller, Field programmable gate array, Industrial Processes, Robot arm, Bench-to helicopter.**

## I. INTRODUCTION

Fuzzy Logic has been successfully applied to a large number of control applications. The most commonly used controller is the PID controller, which requires a mathematical model of the system. A fuzzy logic controller provides an alternative to the PID controller. It is a good tool for the control of systems that are difficult to model. The control action in fuzzy logic controllers can be expressed with simple "if-then" rules [1]. Fuzzy controllers are more sufficient than classical controllers because they can cover a much wider range of operating conditions than classical controllers and can operate with noise and disturbances of a different nature. The method used most often to implement a fuzzy controller is to use it as a computer Program on a general purpose computer; higher density programmable logic devices such as Field-Programmable Gate Array (FPGA) can be used to integrate large amounts of logic in a single IC. FPGA provides additional flexibility: they can be used with tighter time-to-market schedules.

FPGA places fixed logic cells on the wafer, and the FPGA designer constructs more complex functions from these cells. The term field programmable highlights the customizing of the IC by the user, rather than by the foundry manufacturing the FPGA. Several researchers discussed the design of hardware fuzzy logic controllers. A number of these works specialized in control application [2], [3], and were aimed at obtained better control responses. Others were concerned with developing general fuzzy logic processors. Their searches were concerned with using new techniques in fuzzy algorithms, to get higher processing speed versus low utilization of chip resource [4], [5].

Manuscript received August 20, 2009. This work was supported in part by the Department of Electrical & Electronic Engineering, Faculty of Engineering, University Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia. The paper title is "Analysis and Performance Evaluation of PD-like Fuzzy Logic Controller Design based on Matlab and FPGA"

The corresponding author, Zeyad Assi Obaid, is with the Department of Electrical & Electronic Engineering, Faculty of Engineering, University Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia. (Phone: +603-8656 7121, Fax: +6086567099,
E-mail: eng.alhamdany@yahoo.com.)
Nasri Sulaiman, M. H. Marhaban and M. N. Hamidon, are with the Department of Electrical & Electronic Engineering, Faculty of Engineering, University Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia. E-mails: (Nasri,Hamiruce,Mnh)@eng.upm.edu.my, Tel: +603-8656 7121, Fax: +6086567099.

## II. PD CONTROLLER

The D mode is used when the prediction of the error can improve control or when it is necessary to stabilize the system. From the frequency characteristic of the D element it can be seen that it has a phase lead of 90°. Thus, the D element will move the frequency characteristic of the open loop Go (jw) further away from the critical point (-1, j0) [6]. Often, the derivative is not taken from the error signal but from the system output variable. This is done to avoid the effects of the sudden change of the reference input that will cause a sudden change in the value of error signal [6]. A sudden change in error signal will cause a sudden change in control output. This can be avoided by a suitable design of the D mode to be proportional to the change of the output variable y(t). If there is a measuring noise present in y(t) it will amplify this noise [6].

Noise is usually a higher frequency signal, so a good remedy for the noise problem is the use of a low-pass filter in the derivative channel that will insure that the derivative action is only in the frequency band of interest and will diminish the negative effect of the D mode on the signal noise [6]. In the case of D, the PID controllers control signal is proportional not only to the rate of change of process variable

(Advance online publication: 13 May 2010)

but also to the acceleration of the change of process variable. However, these controllers can only be used if the process has good filtering characteristics, (large inertia) since double derivation greatly amplifies noise. When dealing with systems with transport delay it is also important to have a good error prediction. However, the D mode will not be able to give a reliable prediction in the case of transport delay, so in those cases one should use the Otto-Smith predictor (controller), and not the PID controller. If the Otto-Smith predictor is not available it is better to use the PI controller [6].

## III. Fuzzy Vs. Conventional Control

In order to design a conventional controller for controlling a physical system, the mathematical model of the system is needed. A common form of the system model is differential equations for continuous-time systems or difference equations for discrete-time systems. Strictly speaking, all the physical systems in existence are nonlinear. Unless physical insight and the laws of physics can be applied, establishing an accurate nonlinear model using measurement data and system identification methods is difficult in practice. Even if a relatively accurate model of a dynamic system can be developed, it is often too complex to use in controller development, especially for many conventional control design procedures that require restrictive assumptions for the plant (e.g., linearity) [10], [11].

As an alternative, fuzzy control provides a formal methodology for representing, and implementing a human's heuristic knowledge about how to control a system, which may provide a new paradigm for nonlinear systems. The fuzzy controller is unique in its ability to utilize both qualitative and quantitative information. Qualitative information is gathered not only from the expert operator strategy, but also from the common knowledge [10], [11].

Although much of the opposition to fuzzy logic is based on misconceptions, fuzzy control is not a cure-all. Fuzzy control should not be employed if the system to be controlled is linear, regardless of the availability of its model. PID control and various other types of linear controllers can effectively solve the control problem with significantly less effort, time, and cost. In summary, PID control should be tried first whenever possible [10]. The benefits of fuzzy controllers can be summarized as follows:

1) Fuzzy controllers are more robust than PID controllers because they can cover a much wider range of operating conditions than PID, and can operate with noise and disturbances of a different nature.
2) Developing a fuzzy controller is cheaper than developing a model-based or other controller to do the same thing.
3) Fuzzy controllers are customizable, and it is easier to understand and modify their rule, which not only uses a human operator's strategy, but is also expressed in natural linguistic terms.
4) It is easy to learn how fuzzy controllers operate and how to design and apply them to a concrete application.

It is also worth noting that fuzzy logic can be blended with conventional control techniques. This means that fuzzy systems do not necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation [12], [13].

## IV. Fuzzy Logic Controller

Fuzzy logic has rapidly become one of the most successful of today's technologies for developing sophisticated control systems. With its aid, complex requirements may be implemented in amazingly simple, easily maintained, and inexpensive controllers [14]. Fuzzy control only uses a small portion of the fuzzy mathematics that is available. This portion is also mathematically quite simple and is conceptually easy to understand. In this chapter, we introduce some essential concepts, terminology, and arithmetic of fuzzy sets and fuzzy logic. The fuzzy controller, (as explained in Fig. 1), have four main components:

1) The Rule-Base holds the knowledge, in the form of a set of rules, of how best to control the system.
2) The Inference Mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be.
3) The Fuzzification Interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base.
4) The Defuzzification Interface converts the conclusions reached by the inference mechanism into the inputs to the plant [11], [15].

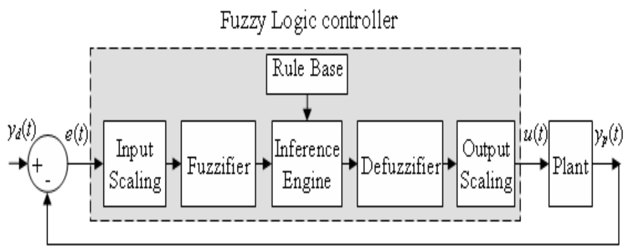



Fig. 1: Structure of fuzzy logic controller with unity feedback control system.

## V. The Matlab-Based PD-like Fuzzy Logic Controller

Generally, this controller accepts two input signals, error and change of error (e(s) and Δe(s)). These inputs represent the PD gain inputs, the error signal is obtained by subtracting the output of the plant (Yp) from the desired output (Yd); the change of error is obtained by multiplying the delay signals with the error signal and then subtracting this signal from the original error signal; the controller delivers the control action signal as output. Groups of five triangular membership functions for inputs/outputs variables, and rule table of 25 rules were used in this design; these groups are shown in Fig. 2 and Fig. 3.

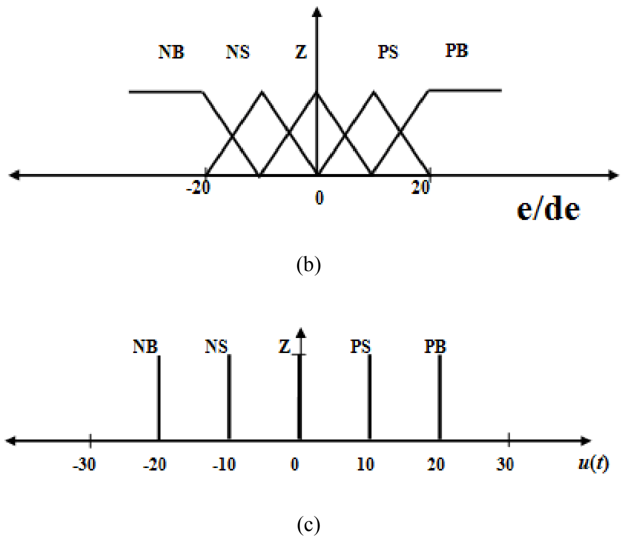



Fig. 2: Inputs/output Member ship functions, (a) inputs, (b) output.

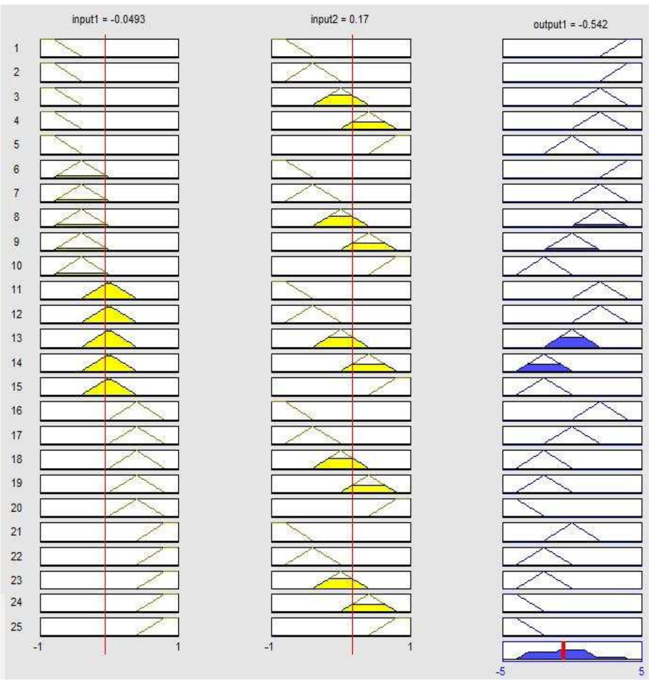



Fig. 3: Fuzzy Rules.

The fuzzy inference was designed using fuzzy toolbox provided in MATLAB, and the same design coded in MATLAB file (.m) in order to make a comparison with other FPGA-based designs. Fig. 4 shows the layout of the fuzzy inference in the PD fuzzy logic controller.

For any fuzzy logic controller design, it is necessary to check the surfaces between the proposed membership function and the control action in order to make sure of the rounding process inside the fuzzy system. Fig. 5 shows the control surface between inputs/output variables using the proposed membership functions.

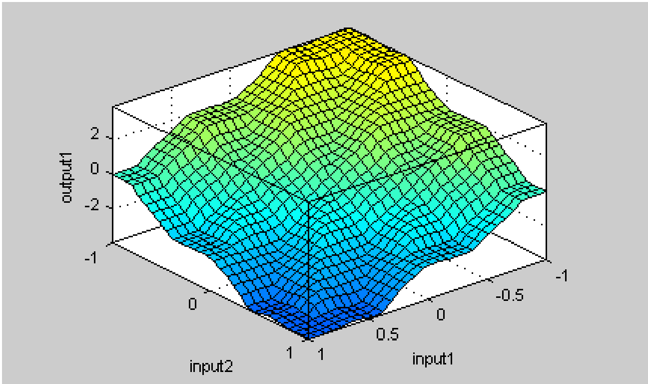



Fig. 5: Control surface between inputs/output variable using fuzzy toolbox.

## VI. The FPGA-Based PD-Like Fuzzy Logic Controller

Generally, this controller accepts the output of the plant (Yp) and the desired output (Yd), both as digital signals, and delivers the digital control action signal as output. The design also accepts digital signals that represent the gain coefficients needed by the controller (proportional gain Kp and derivative gain Kd). Fig. 6 shows the layout of the proposed controller with a unity feedback control system. For the purpose of simulation during the design we used five triangular membership functions for input variable, five singleton membership functions for output variable, and rule table of 25 rules. This group is shown below in Fig. 7 and Table 1.

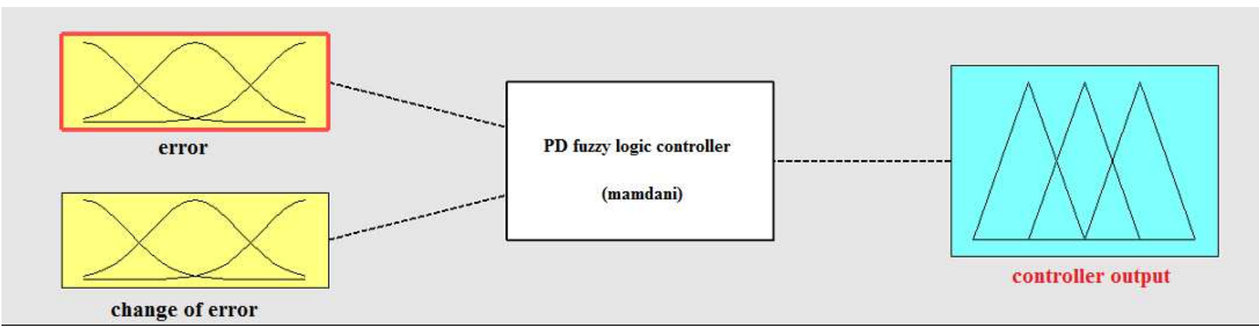



Fig. 4: Mamdani Type fuzzy inference in the proposed PD fuzzy logic controller.

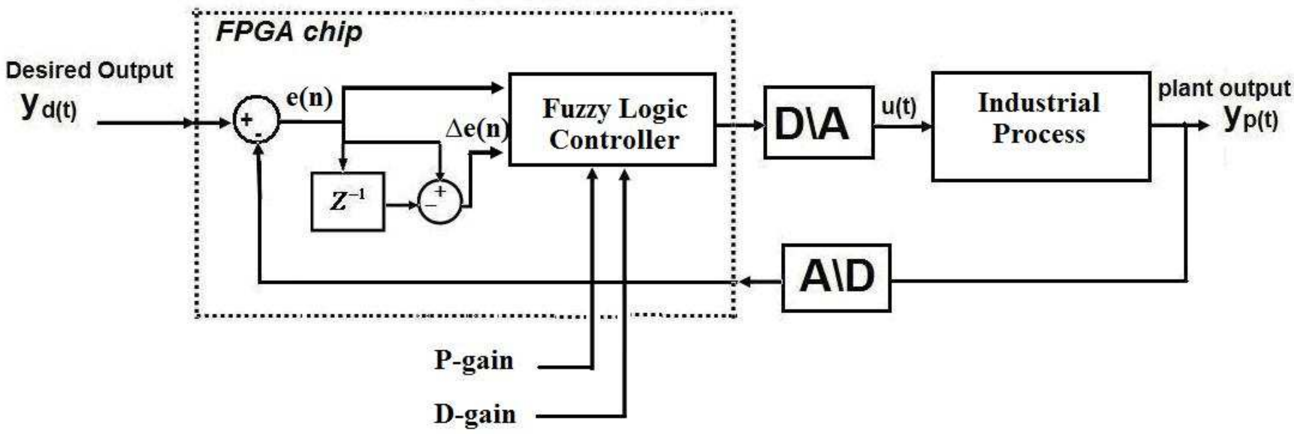



Fig. 6: Layout representation of the FPGA-based PD-like fuzzy logic controller with unity feedback control system.

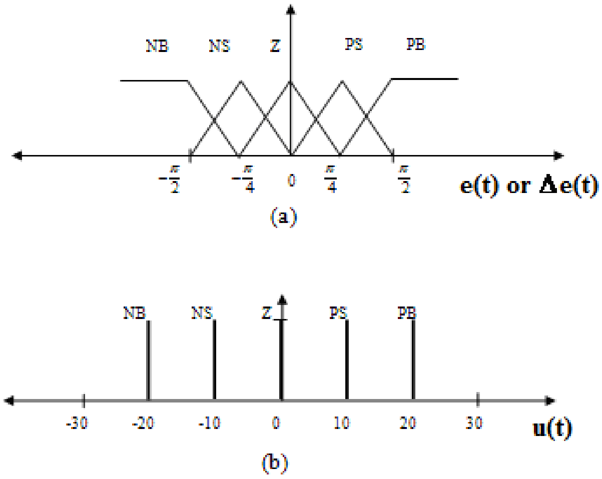



Fig. 7: (a) Inputs fuzzy sets, (b) Output fuzzy sets

Table 1: Fuzzy rules used in FPGA-based Design.

| | | e | | | | |
|---|---|---|---|---|---|---|
| | | NB | NS | Z | PS | PS |
| de | NB | PB | PB | PB | PS | Z |
| | NS | NS | PB | PS | Z | NS |
| | Z | PB | PS | Z | NS | NB |
| | PS | PS | Z | NS | NB | NB |
| | PB | Z | NS | NB | NB | NB |

## VII. Simulation Environments

In FPGA-based design, the *Altera Quartus II 9.0* program was used to get the compilation and timing test results as well as the synthesized design. For the simulation results, the Quartus simulator with *ModelSim 9.0* simulation program was used with the FPGA-based Design. In Matlab-based design, a Simulink design was used to represent the design layout and then the simulation graphs were plotted. The same Matlab-based design was coded using M-file. In order to make the comparison with the FPGA-based design another M-file was used to plot the data collected from the ModelSim program to make the comparison between both designs. This comparison is necessary to analyze and evaluate the performance of both types of controller.

## VIII. Simulation Results of Matlab-based Design

For the simulation results with this design, mathematical models of three different plants were used to generate the results. The first system represents a mathematical model of – PH-control, temperature control or level control. The second model represents the position control in the AC motor and the last model represents the robot arm.

### *A. Arm Control of the Non linear Robot*

Many researchers discuss the behavior of the robotic arm. In order to get better control solutions, this paper presents the design of the PD fuzzy logic controller to control the robot arm. The robot arm model consists of four popular elements, which are controlled in our design. These elements are, *Velocity*, *Position*, *Friction* and *Gravity function*. Many cases were used in this paper by changing the value of these elements in order to get better control responses. Equation (1) shows the deferential equation, which represents the robot arm model; [7].

$$\ddot{y} = u - 2\dot{y} - 10y * \sin(u) \qquad (1)$$

The proposed PD fuzzy logic controller with unity feedback control systems was used to control the robot arm. This workbench design was used to generate the simulation results by changing the elements of the robot arm, in order to get better control responses and also to analyze and evaluate the performance of this controller. Fig. 8 shows the layout of the proposed controller with unity feedback control system.

The robot arm model consists of four popular elements, which were controlled in our design. These elements are, *Velocity*, *Position*, *Friction* and *Gravity function*. Many cases were used to generate the simulation results. These cases were made by changing the value of the four elements inside the robot arm. These simulation results are shown in Fig. 9 to Fig. 12. From the simulation results, it seems that all the responses still have an error in their plant output, as the PD parameters affect on the transient response not the steady state response, and the error belongs to the steady state response. Therefore, all changes were made with the parameters effect on transient response (overshoot), raising the time and settling time only.

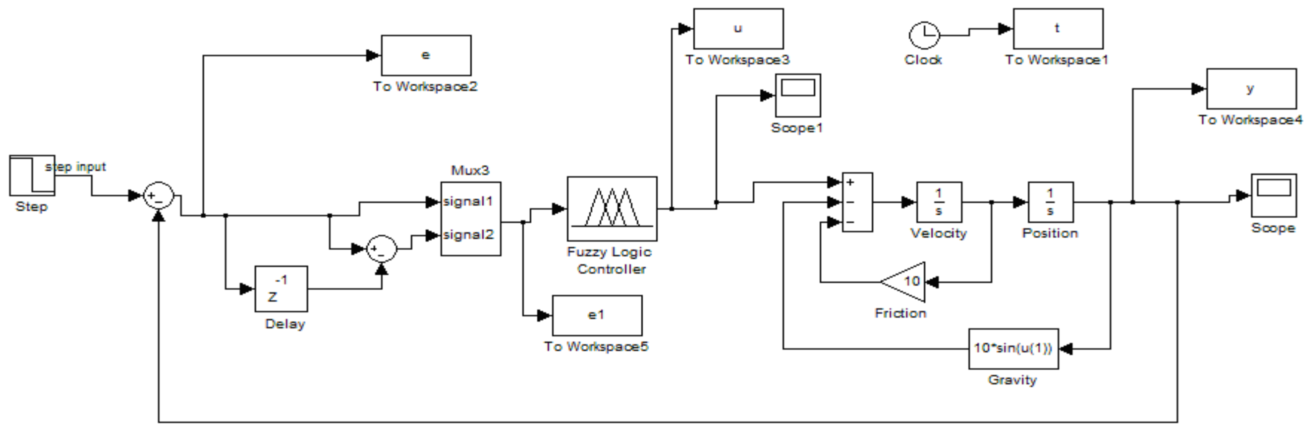
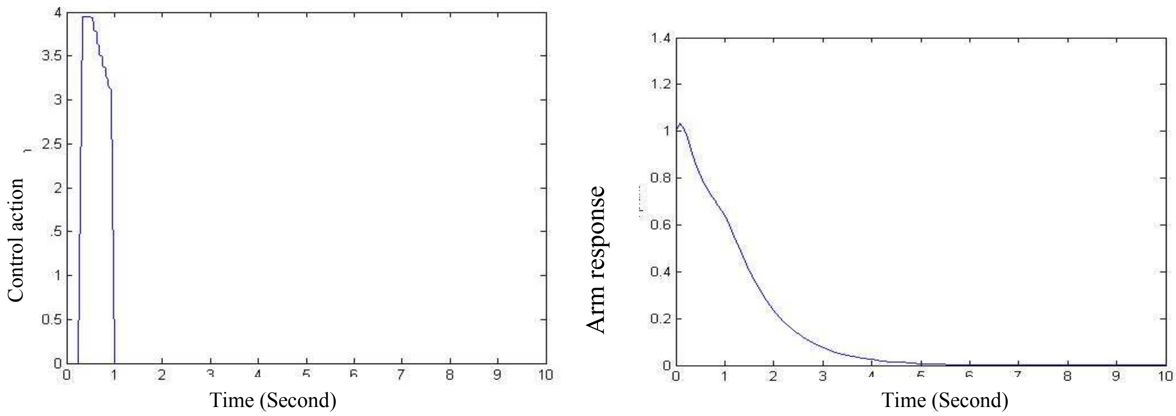
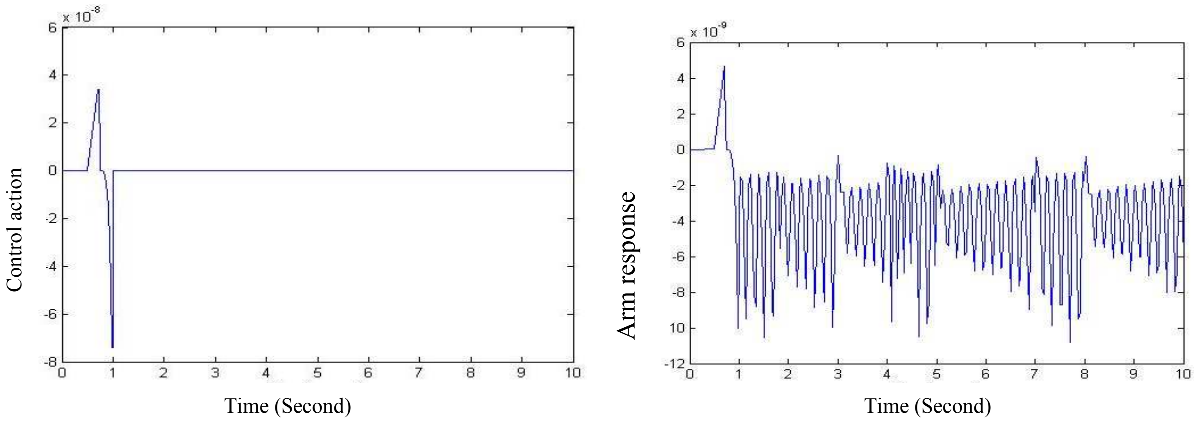



Fig. 8: MATLAB Simulink of the proposed controller with unity feedback control system to control the robot arm.



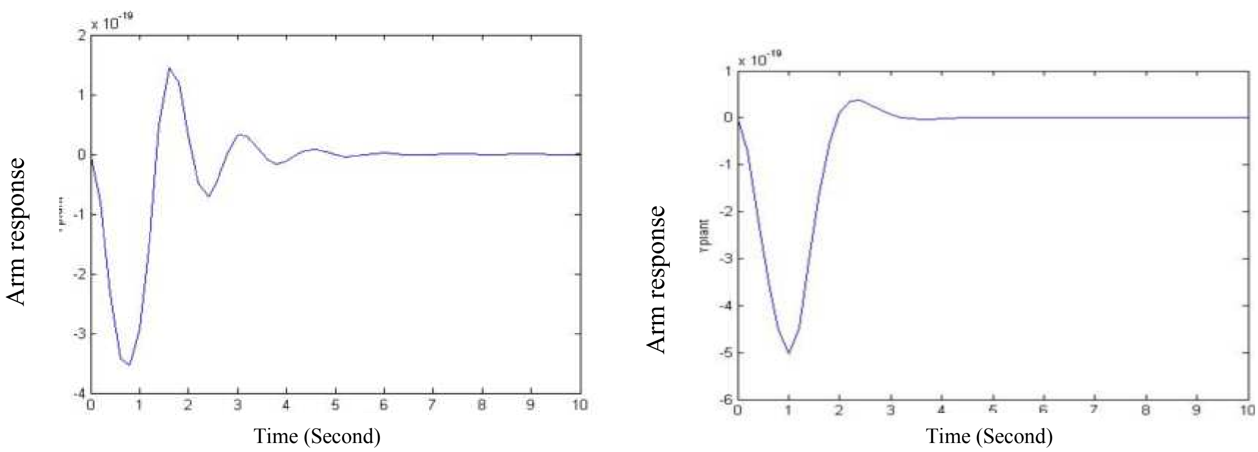

(a) (b)

Fig. 9: Initial Value, Velocity=1, Position=1, Friction=10, (a) Control action, (b) Plant output.



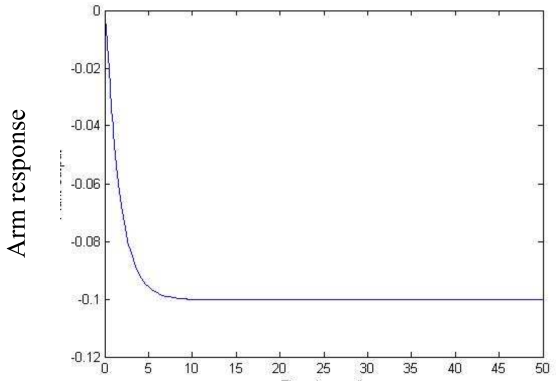

(a) (b)

Fig. 10: Initial Value, Velocity=0, Position=0, Friction=100, (a) Control action, (b) Plant output.

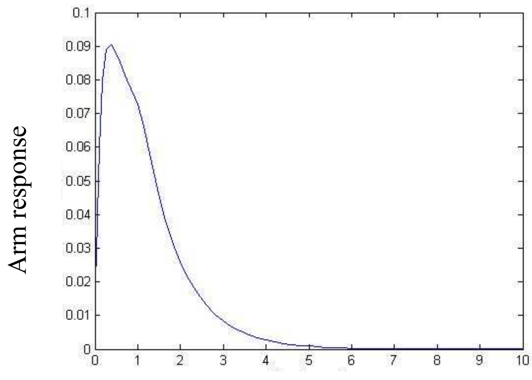



(a) (b)

Fig. 11: Different cases of Plant output responses, (a) Friction= 2, Initial Value, Velocity=0, Position=0, (b) Friction= 4, Initial Value, Velocity=0,




(a)




(b)

Fig. 12: Different cases of Plant output responses, (a) Friction= 10, Initial Value, Velocity=0, Position=0, (b) Friction= 10, Initial Value, Velocity=1, Position=0.

### *B. Bench-top Helicopter Model*

The bench-top helicopter is shown in Fig. 13. It is a laboratory scale plant with 3 Degrees of Freedom (3DOF), roll angle *1*, pitch angle *2*, and yaw angle *3*. Each one is measured by an absolute encoder [8], [9].

Two electrical DC motors are attached to the helicopter body, making two propellers turn. The total force *F* caused by aerodynamic makes the total system turn around an angle measured by an encoder. A counter weight of mass M helps the propellers lift the body. The total non-linear model obtained for the helicopter model can be simplified by linearising around the operational point □0=0. It yields a second order transfer function between the pitch angle *2* and the motor signal [8].

$$P(s) = \frac{\theta(s)}{U(s)} = \frac{k\, w_n^{\ 2}}{s^2 + 2\zeta w_n s + w_n^{\ 2}} \qquad (2)$$

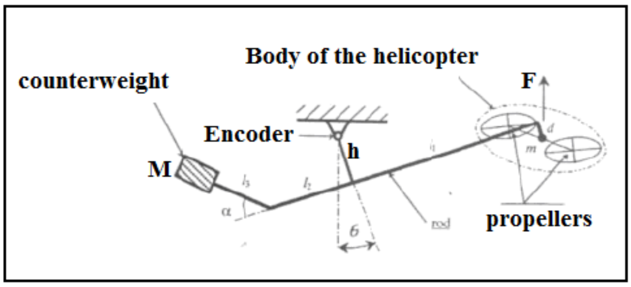



Fig. 13: A laboratory scale bench-top helicopter [8].

Using system identification techniques from experimental data the parameter variations obtained are: $k$ = [0.01, 0.099]; $\zeta$ = [0.1 0.16] and $w_n$ = [0.55, 0.58], [8].

The simulation results were generated by using different plat's parameters by changing these values. Fig. 14 shows the system design by using Matlab Simulink with unity feedback control system. Table (2) shows the parameters used to generate the simulation results using equation (2). These values are necessary to analyze our design. Fig. 15 to Fig. 17 shows the simulation results. This system is a type zero system, which always has an error in its steady state response. PD can only affect the transient response. This means that it can only reduce the overshoot, as these systems always need an integral value in order to reduce the error.

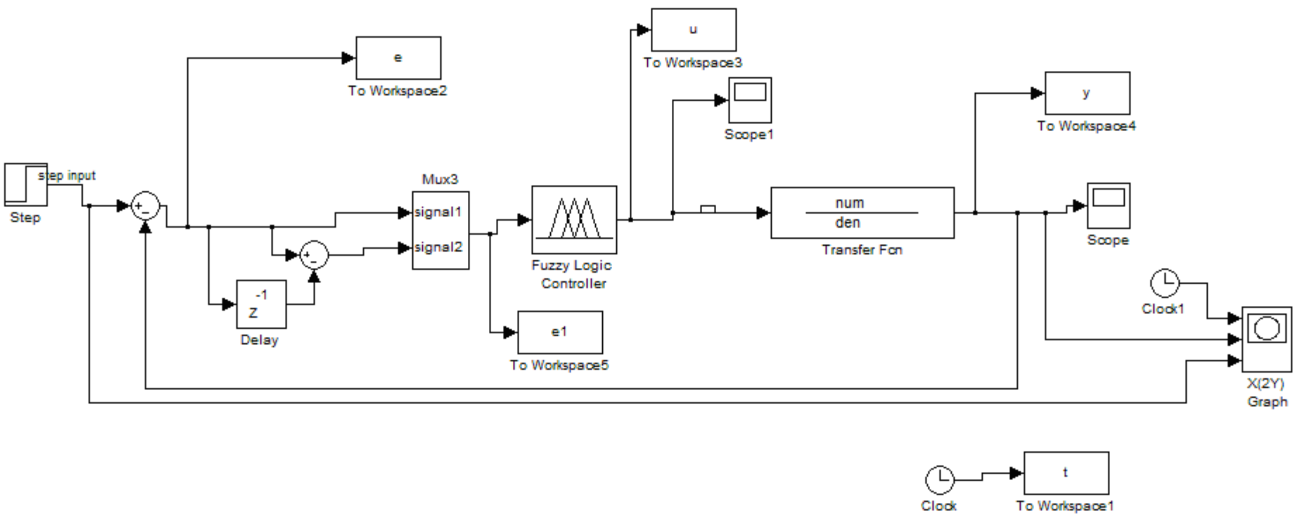
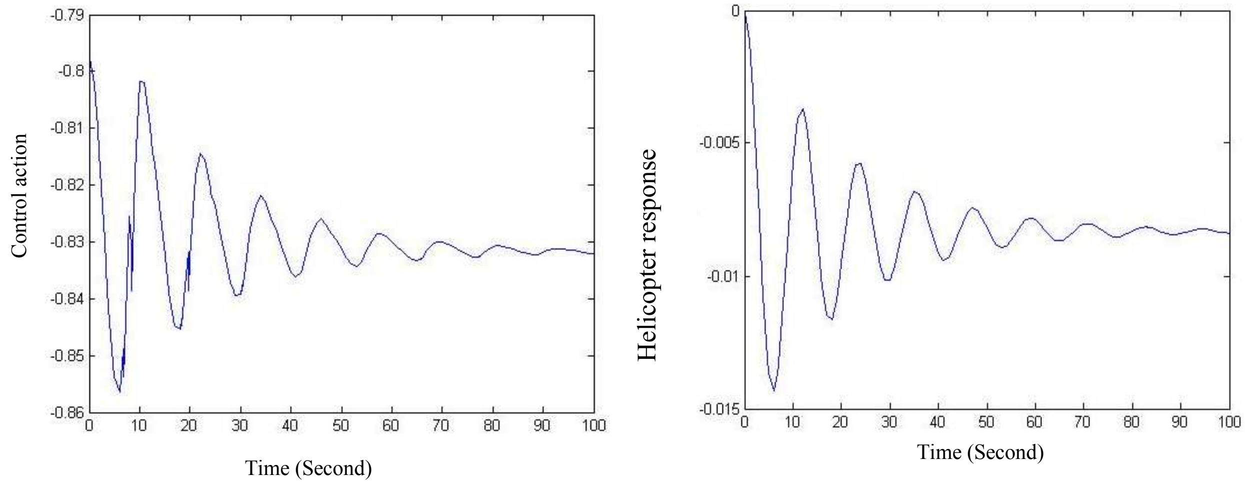



Fig. 14: MATLAB Simulink of the controller with unity feedback control to control the bench-top helicopter.

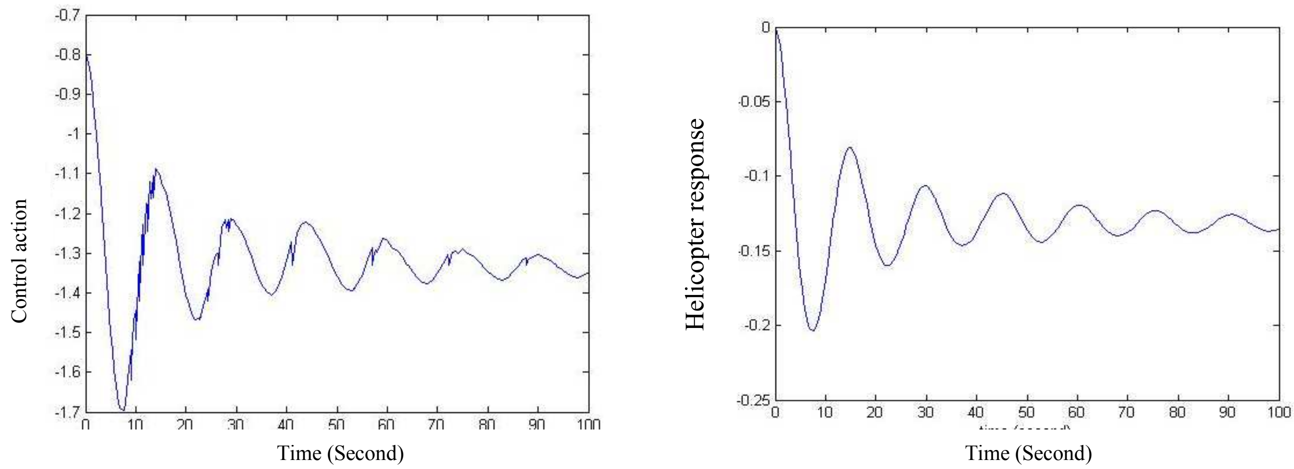
Table 2: Parameters Values used to generates the simulation results (modified from [8]).

| System | k | $\zeta$ | $w_n$ |
|---|---|---|---|
| Case 1 | 0.01 | 0.1 | 0.55 |
| Case 2 | 0.099 | 0.16 | 0.58 |
| Case 3 | 0.099 | 0.2 | 0.58 |



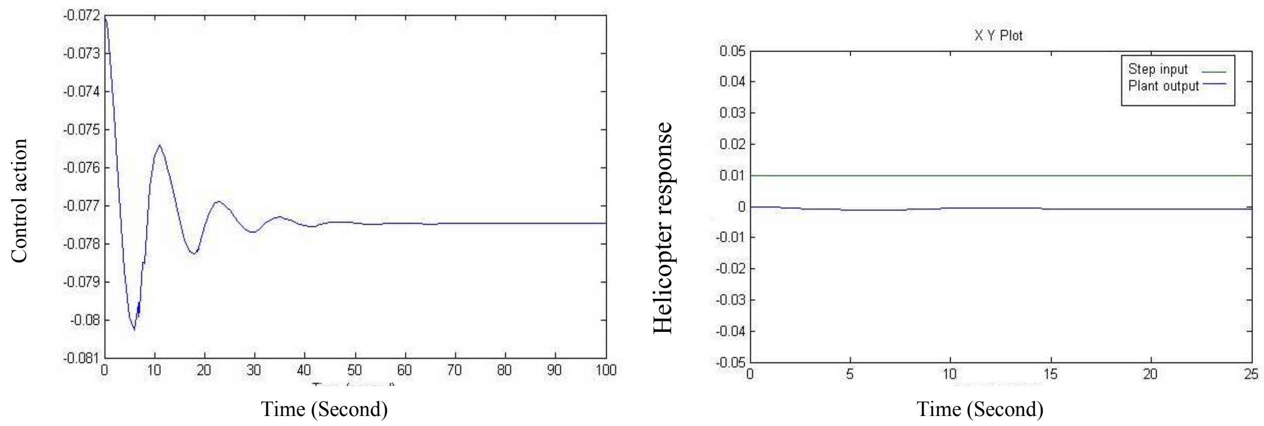

(a) (b)

Fig. 15: Case 1, PD-like Fuzzy logic controller with bench-top helicopter, (a) Control Action Output, (b) Plant Output.




(a) (b)

Fig. 16: Case 2, PD-like Fuzzy logic controller with bench-top helicopter, (a) Control Action Output, (b) Plant Output.

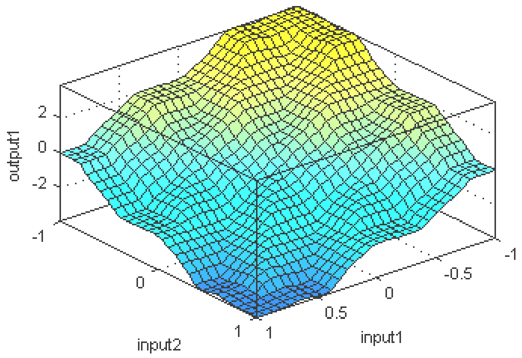



(a) (b)

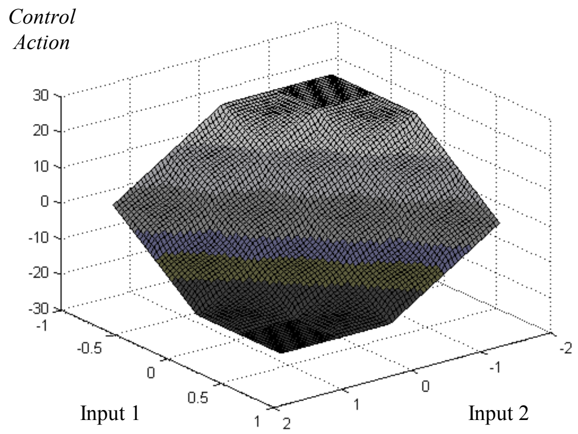
Fig. 17: Case 3, PD-like Fuzzy logic controller with bench-top helicopter, (a) Control Action Output, (b) Plant Output.

## IX. SIMULATION RESULTS OF FPGA-BASED DESIGN

At first, a test is performed to make sure that the fuzzy inference system used inside the *FPGA-based* design is working properly. This test involves generating the control surface using fuzzy sets and rule table shown in Fig. 7 and Table 1. Fig. 18-a shows the surface of the fuzzy inference, which is implemented using VHDL. Fig. 18-b shows the Control surface of the typical fuzzy logic controller using MATLAB.




(a)




(b)

Fig. 18: a) Control surface of the PD fuzzy logic controller using VHDL. b) Control surface of the PD fuzzy logic controller using *MATLAB*.

### A. Control of Industrial Processes

Many of the industrial processes, such as temperature, pressure, pH, and fluid-level controls, can be approximated by first order models (eq. 3 and 4) [16]. This model is used to generate the simulation results; it is designed with Non-synthesizable VHDL code for simulation purposes using *ModelSim* as shown in Fig. 6. It uses this simulation data to represent the graphs.

$$CS_1(s) = \frac{1}{s+1} \tag{3}$$

$$CS_1(z) = \frac{0.09516}{Z - 0.9048} \quad , T = 0.1 \tag{4}$$

Delayed first order plant:

$$G_2(z) = z^{-2} \times G_1(z) \tag{5}$$

$$CS_2(z) = Z^{-2} \times \left( \frac{0.09516}{Z - 0.9048} \right) \quad , T = 0.1 \tag{6}$$

The FPGA-based PD-like fuzzy logic controller is used in unity feedback control systems, (as shown in Fig. 6), and subjected to 0.5 step input. The plant responses of the proposed FPGA-based is plotted using the simulation data from the ModelSim program. Fig. 19 shows the synthesized design results of the PD fuzzy logic controller using VHDL. From the simulation results, the controller is able to produce a response in 0.3 µs (300000 ps). Fig. 20 shows the simulation results obtained form ModelSim. These results were generated by using: Non-synthesizable VHDL code. This code is shown in Fig. 21.

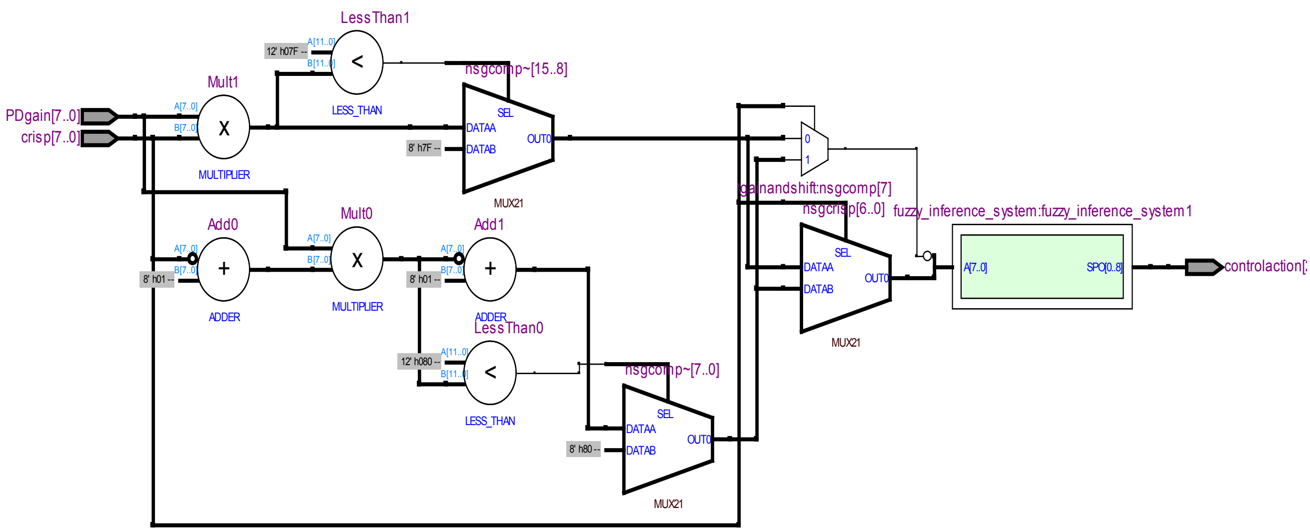
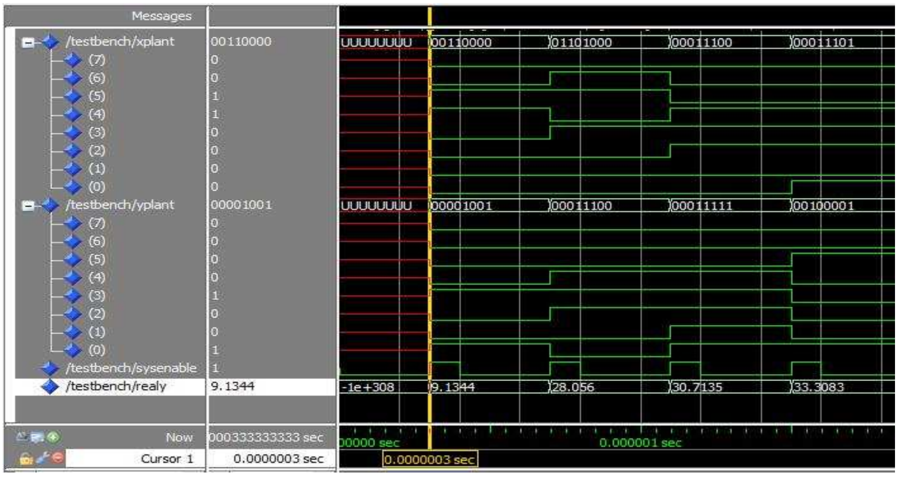
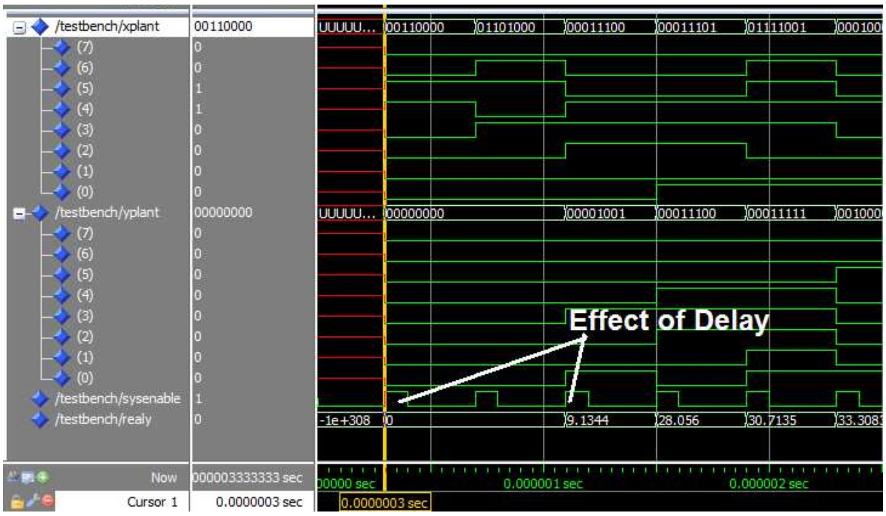



Fig. 19: Synthesized design results of the PD fuzzy logic controller.




(a)




(b)

Fig. 20: Simulation waveform results of the PD fuzzy logic controller, (a) with G1, (b) with G2.

```
architecture firstorder of plant is

begin
process

 variable ry1:real:=0.0;
variable rx1:real:=0.0;
begin
wait until sysenable'event and sysenable='1';

rx1:=real(conv_integer(xplant));
ry1:= (rx1*  0.1903  +  ry1* 0.9048);
if ry1>127.0 then
yplant<="01111111";

elsif ry1<(-128.0) then
yplant<="10000000";

else
yplant<=conv_std_logic_vector(integer(ry1),8);
end if;

realy<=ry1;
end process;

end firstorder;
```

Fig. 21: Non-synthesizable VHDL code.

## X. Discussion and Conclusion

This paper presents an analysis and performance evaluation of PD-like fuzzy logic controller design by using Matlab and FPGA. The simulation results for the first type prove that the PD fuzzy logic controller followed the specifications that belong to the classical PD controller.

By applying the first type with the robot arm, the better control responses were dependent on the elements of this model. The friction was the most effected element on these responses. It ranged from 10 to 15 and had 0.02 errors and zero overshot. From these simulation results, it seems that all responses still have an error in their plant output, as the PD parameters affect the transient response not the steady state response, and the error belongs to the steady state response.

By applying the first type with the mathematical model of the bench-top helicopter, the better responses with the bench-top helicopter have 0.01 errors and 0.001 overshot. This system is a type zero system, and these systems always have an error in their steady state response. PD can only affect the transient response. This means that it can only reduce the overshoot. As these systems always need an integral value in order to reduce the error, the better responses have been obtained by using Case 3 in the helicopter model.

The second type, with the mathematical model, which represents industrial processes, such as temperature, pressure, pH, and fluid-level controls with unity feedback control systems and subjected to 0.2 step input has been used to generate the simulation results of this type. This FPGA-based controller is able to produce a fast response ranging from 0.3 μs (300000 ps), even with time delay added to the plant model. The FPGA-based type has control responses faster than the Matlab-based type.

## ACKNOWLEDGMENT

The authors would first like to thank God and all the UPM staff and all friends who provided help related to this work. Finally, heartfelt thanks to our families and home countries.

## REFERENCES

[1] S. Poorani, T.V.S.U. Priya, K.U. Kumar, and S. Renganarayanan, "FPGA Based Fuzzy Logic Controller for Electric Vehicle," *Journal of the Institution of Engineers*, Singapore, Vol. 45 Issue 5 2005.

[2] O. Karasakal, E. Yesl, M. Guzelkaya, I. Eksin, Implementation of a "New Self-Tuning Fuzzy PID Controller on PLC," *Turk J Elec Engin*, Vol.13, NO.2, 2005.

[3] V. Tipsuwanporn, S. Intajag and V. Krongratana, "Fuzzy Logic PID Controller Based on FPGA for Process Control," Proceedings of IEEE International Symposium on Industrial Electronics, Vol. 2, May 4-7, 2004, page(s):1495-1500.

[4] M.Y. Hassan and W.F. Sharif, "Design of FPGA based PID-like Fuzzy Controller for Industrial Applications," *IAENG International Journal of Computer Science*, Vol. 34 Issue 2, November 2007.

[5] Z.A. Obaid, N. Sulaiman and M.N. Hamidon, "Developed Method of FPGA-based Fuzzy Logic Controller Design with the Aid of Conventional PID Algorithm", *Australian Journal of Basics and Applied Science*, Vol. 3(3), P: 2724-2740, 2009.

[6] Z. Vukic and O. Kuljaca, "*Lectures on PID Controllers*," Faculty of Electrical Engineering and Computing, University of Zagreb, Automation and Robotics Research Institute, University of Texas at Arlington, April, 2002.

[7] Matlab Tool Box, "*Reference Control of a Robot Arm*," Provided with Matlab tool box- neural networks, Matlab version 2008.

[8] H. Mansor, A.H. Zaeri, S.B. Mohd. Noor, R. Mohd. K. Raja Ahmad, F. Saleena and H. I. Ali, "Design of QFT Controller for a Bench-top Helicopter," The Second International Conference on Control, Instrumentation and Mechatronic Engineering (CIM09) Malacca, Malaysia, p:19-24, June 2-3, 2009

[9] C. H. Houpis, S. J. Rasmussen, and M. Garcia-Sanz, "*Quantitative Feedback Theory, Fundamentals and Applications*," Taylor & Francis, 2006.

[10] H. Ying, "*Fuzzy Control and Modeling, Analytical Foundations and Applications*," Institute of Electrical and Electronic Engineers Inc., USA, 2000.

[11] K.M. Passino and S. Yurkovich, "*Fuzzy Control*," Addison-Weslwey Longman Inc., USA, 1998.

[12] L. Reznik, "*Fuzzy Controllers*," Newnes, first edition, 1997.

[13] Mathworks, "*Fuzzy Logic Toolbox User's Guide*, Mathworks, Inc., 1999.

[14] E. Cox, "*Fuzzy Fundamentals*," IEEE Spectrum, Vol. 29, Issue 10, October 1992.

[15] J. Huang, "Hybrid Fuzzy PID Controller with Adaptive Genetic Algorithms for the Position Control and Improvement of Magnetic Suspension System," M. Sc. thesis, Mechanical and Electro-Mechanical Engineering, June, 2004. http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search/view_etd?URN=etd-0624104-182807

[16] B.G. Hu, G.K. Mann and R.G. Gosine, "New Methodology for Analytical and Optimal Design of Fuzzy PID Controllers," *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 5, pp. 521-539, October, 1999.

***Eng. Zeyad Assi Obaid*** was born in Dyala, Iraq. He received his B.Sc. Degree in Control and Systems Engineering from the University of Technology, Baghdad, Iraq, in 2006. In 2007, He received Cisco Certified Network Association (CCNA) from (Cisco Computer center), Doha, Qatar. He received Master Degree in Control and Automation Engineering Department form University Putra Malaysia, 2010.

His current research interests include fuzzy control systems with FPGA design applications. He has many Professional Affiliations including, Member of IAENG, Member of ACM, Member of IACSIT, and Member of IGNSS. He has been invited to review papers in the IEEE International Conference on Signal and Image Processing Applications (ICSIPA), and IETE Technical Review (journal) by the Institution of Electronics and Telecommunication Engineers (IETE). He has worked in various companies, including: from 2004 to 2006, working as: Network Administrator in Alkawarizmy Computer Services Company, Baghdad, Iraq. From 2006 to June-2008 working as Control and Computer Engineer in Ezdan Real estate Company (International Constructions and Trading Company), Doha, Qatar. He has attended many Technical Workshops and Seminars and participates in many International Conferences. He has many publications in conference proceedings and journals.

***Dr. Nasri Sulaiman*** was born in Malaysia; he received his B.Sc. Degree in (Elect & Comp.) from University Putra Malaysia, 1994. Master degree from University of Southampton, UK, 1999. PhD from University of Edinburgh, UK, 2007. He is currently a Lecturer in the Department of Electrical and Electronics Engineering, UPM. His current research interests include. Evolvable Hardware (EHW) and Digital Signal Processing (DSP).

***Assoc. Prof. Dr. M. H. Marhaban***. Was born in Malaysia; he received his B.Sc. Degree in (Electrical and Electronic Engineering) from the University of Sanford, UK, 1998. Ph.D. (Electronic Engineering) form University of Surrey, UK, 2003. A-Level, Mara Science College, Kuala Lumpur, 1995. Malaysia Certificate of Education, Mara Junior Science College, Taiping, Perak, Malaysia, 1993. Lower Certificate of Education, Sekolah Menengah Gua Musang, Kelantan, Malaysia, 1991. He is currently a Lecturer in the Department of Electrical and Electronics Engineering, UPM. His current research interests include: Intelligent Control System and Computer Vision.

***Dr. M. N. Hamidon*** was born in Malaysia; he received his B.Sc. Degree from the University of Malaya, Malaysia, 1994. Master degree from University Kebangsaan, Malaysia, 2001. PhD from the University of Southampton, UK, 2005. He is currently a Lecturer in the Department of Electrical and Electronics Engineering, UPM. His current research interests include. Microelectronics (Sensor Technology), MEMS, Wireless System Devices Fabrication and Packaging.