Design of FPGA Based P/PI/PD/PID Controller for Industrial Applications

Dr. Mohammed Yousif Hassan
Control Systems Engineering Department
University of Technology, Baghdad, Iraq

Asst. Lect. Muayad Sadik Croock
Computer Eng. and Information Technology Dept.
University of Technology, Baghdad, Iraq

Asst. Lect. Sahar Salman Mahmood
Applied Sciences Department
University of Technology, Baghdad, Iraq

Abstract

Digital PID controllers are widely used in industrial applications. They are used in autopilots, ship steering, and speed control of machines…etc. A digital computer with an interfacing device is used to implement such types of controllers.

In this paper, the design of digital controller using FPGA is described. The design provides a minimization in the hardware design and reduction in the cost as compared with the digital control implemented using digital computers.

The structure of the controller can be selected using 2 bit input binary value. The type of the structures is P, PI, PD or PID. The parameters of the controller are tuned from the outside of the FPGA controller. In order to test the design, a discrete linear 1st order model is implemented inside the FPGA. Also, the closed loop system was simulated using MATLAB/SIMULINK program. A comparison between the simulated and designed systems shows that the results are very acceptable and close to each other.

الخلاصـــــة

يستخدم المسيطر الرقمي من نوع المتناسب- المتكامل - المتماثل بصوره واسعة في التطبيقات الصناعية، حيث يستخدم في الطيار الآلي ومسيطر مقود السفينة وفي السيطرة على سرعة المكائن . لغرض تنفيذ مثل هذا النوع من المسيطرات يتم استخدام حاسوب رقمي مرتبط مع أجهزة تعشيق.

يتناول هذا البحث عملية تصميم مسيطر رقمي باستخدام تقنية مصفوفة بوابات المجال القابلة للبرمجة، حيث يمنح هذا التصميم تقليلا في حجم الكتل المستخدمة وكذلك تقليلا في كلفة التنفيذ مقارنة مع المسيطر الرقمي المنفذ باستخدام حاسوب رقمي .

يمكن في هذا التصميم اختيار هيكل المسيطر باستخدام مدخلين ثنائيين بمقدار بت واحد لكل منهما حيث يمكن اختيار عدة أنواع من الهياكل تشمل أما مسيطر متناسب أو مسيطر متناسب-متكامل أو مسيطر متناسب-متماثل أو مسيطر متناسب-متكامل-متماثل. لغرض فحص تصميم المسيطر تمت إضافة مود يل رياضي من النوع المتقطع بنظام خطي من الدرجة الأولى يمكن توصيله مع المسيطر. كذلك تم تمثيل المنظومة المغلقة بطريقة رسم الموديل في برنامج مختبر المصفوفات. وقد تم عمل مقارنه بين المنظومة المغلقة في كلا الحالتين وبكل أنواع الهياكل للمسيطر ووجد أن النتائج كانت مقبولة ومتقاربة مع بعضها.

1. Introduction

Proportional plus Integral plus Derivative (PID) is the most used controller type in industry, and is available as a stock item. However, its use is so diversified that the control engineer must tune the PID values according to specific needs.

Studies have guided industry by providing quantitative data for tuning PID controllers for given process, operational conditions, and performance criteria. Unfortunately, these studies considered only one kind of functional PID control structure, the most popular one [1,2]. The tuning of PID controller for industrial plants can frequently be difficult and time consuming, even if expert systems are use to automate the tuning process. All tuning processes are focused on finding the parameters of the traditional cascaded PID structure.

PID controllers are used in most industrial applications, like the firing angle calculations in DC drives, ship steering, and autopilots of aircrafts, …etc [3]. Most manufacturers of PID algorithms fit under one of three major classifications: interactive, non-interactive and parallel [2].

Many researchers deal with the design and implementation of digital PID controllers using either a microprocessor or a microcontroller [3,4,5]. The tuning of PID controller using auto-tuning was also proposed. One of the new techniques is using Genetic Algorithm (GA) [6].

2. Field Programmable Gate Array (FPGA)

A Field Programmable Gate Array (FPGA) is a programmable logic device that supports implementation of relatively large logic circuit. FPGAs are quite different from Simple Programmable Logic Devices (SPLDs) and Complex Programmable logic devices (CPLDs) because FPGAs don't contain AND or OR planes. Instead, FPGAs provide logic blocks for implementation of required functions [7].

The general structure of an FPGA contain three main types of resources logic blocks, input/output blocks connecting to the pins of the package, and interconnection wires and switches.

The logic blocks are arranged in a two-dimensional array and the interconnection wires are organized as horizontal and vertical (routing channels) between rows and columns of logic block. The routing channels contain wires and programmable switches that allow the logic blocks to be interconnected in many ways; each switch contains two terminals. The input terminal is connected to the logic block and output terminal to the interconnection wires. Programmable connection also exists between the input output blocks and the interconnection wires. The actual number of programmable switches and wires in an FPGA varies in commercially available chips [8].

Different companies were interested with this type of manufacturers. 'Xilinx' is one of the major manufactures of FPGAs defines them as high density Application Specific Integrated Circuits (ASICs) combining the logic integration of custom Very Large Scale Integration (VLSI) with the time of market and cost advantages of standard products.

Another major producer "Lucent" also stresses versatility of the FPGA in its publications and of time to market benefit. In the beginning FPGAs where mostly looked at as being of importance for raped prototyping of circuit which would later be hardwired and thus of being a design tool to aid profitability rather than being of use directly in applications. That is changing as chips become available with higher gate counts on tools and faster routing and placing tools becoming available [9].

FPGA can be used to implement logic circuits of more than a few hundred thousand equivalent gates in size [10]. Each logic block in an FPGA typically has a small number of inputs and one output. A number of FPGA products are on the market, featuring different types of logic blocks. The most commonly used logic block is a Lookup Table (LUT), which contains storage cells that are used to implement a small logic function, each cell is capable of holding a single logic value, either 0 or 1. The stored value produced as the output of the storage cell. LUTs of various sizes may be created, where the size is defined by the number of inputs.

The software environments that represent the software package of FPGA have two main types. Firstly, Foundation series and secondly Navigator series.

The design of logic circuit could be done with above software by two ways, either schematically or Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL).

2-1 Schematic Design

The schematic design can be done using schematic editor of any type of two software's environment, where they have same idea of use. The schematic editor has a library especially for each series of FPGA and has an option to make your device that you want to design easily. All the types of connection wires then can be tested using the simulation of work of the software. After design and simulation were tested, the design then is downloaded to the specified FPGA [7].

2-2 VHDL Design

This program was sponsored by the Development of Defense (DOD) with the goal of developing a new generation of high-speed integration circuits. When the life cycle cost of the hardware systems be very high the need for standardized representation of digital system become apparent. So, the first version of language was released in 1987 [11]. The language was subsequently transferred to the IEEE for standardization after which representatives from industry, government, and academe where involved in its further development [12].

VHDL is a language for describing digital systems. Such descriptions can be used by a simulator for simulating the behavior of the system without having to actually construct the system. Alternative synthesis compiler can utilize such a description for creating description of the digital hardware for implementing the system [13].

3. Implementation of P/PI/PD/PID controller

The PID controller digital circuit was implemented using FPGA. It is designed with 2's complement number representation. The results are converted into signed numbers with the N-12 floating number representation method. The input data to the system that comes from the ADC's are converted into 2's complement value by complementing the signed bit. The main block diagram of the controller is shown in **Fig.(1)**. This block shows the different inputs into the system. Also, it has one output, which represents the PID output, (PID_OUT) 12bits. The value is produced from the adder of the controller after approximating the internal floating value to the nearest rounding integer. Mclk (1 bit output), is the main system clock that must be equal to 32 times the sampling frequency. The other input values and signals are:

DC, 1bit:	*is the D controller latch enable.*
IC, 1bit:	*is the I controller latch enable.*
KD, 12 bits:	*is the derivative (D) gain constant, (0.0-16.9999).*
KI, 12 bits:	*is the integral gain (I) constant, (0.0-16.9999).*
KP, 12 bits:	*is the proportional gain constant, (0.0-16.9999).*
SP, 8 bits:	*is the set-point value.*
TS, 8 bits:	*is the sampling time, (1.00-1000.00 msec).*
Y, 8 bits:	*is the feedback value from the controlled system (test model).*

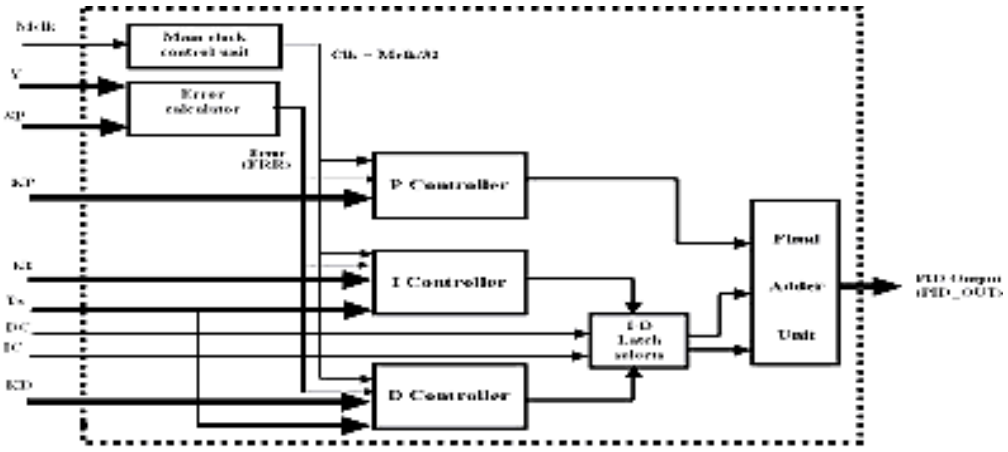


Figure (1) Controller main block diagram

The error that is input to the controller is represented with a floating point value within a range of 16 bits. While the D controller and the I controller outputs are represented with a floating point value within a range of 20 bits.

However, I and D controllers can be activated using two input bits, (DC and IC), these two bits controlled the work of each (I,D) controller as shown in **Table (1)**.

Table (1) Selection of controller type

Controller type	DC	IC
P	0	0
PI	0	1
PD	1	0
PID	1	1

3-1 Clock Control Part

This part is responsible of the clock generation (CLK). There is one of a sequence dividers in the digital circuit that needs a (32) clock to give the results. This part contains five (D flip-flops) that are connected serially to divide the main clock by (32) to make synchronization between all components. This part is shown in **Fig.(2)**.

Figure (2) Clock control part diagram

3-2 Digital PID Controller Block Part

The PID part contains the digital circuits of the (P) controller, (I) controller, and (D) controller, shown in **Fig.(3)**. The controller equations are implemented using difference equation technique.

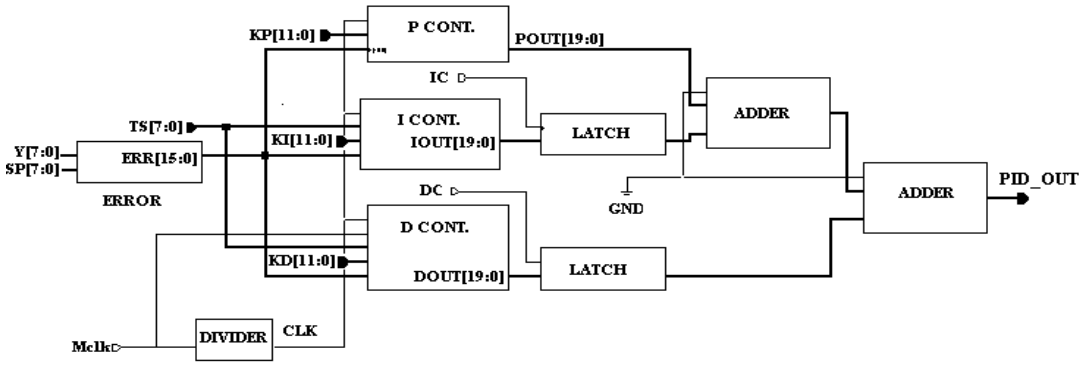


Figure (3) The PID block circuit diagram

3-2-1 Error Calculation

The error is calculated by subtracting the set-point value (SP) from the output of the plant value (Y):

$$\mathbf{ERR(K) = SP - Y(K)} \quad \text{.. (1)}$$

where: ERR(k) is the error between the set-point value and the actual value at kth sampling time and Y(k) is the output of the plant at kth sampling time.

The block diagram of the error calculation block is shown in **Fig.(4)**. The block consists of a 2's complement block to complement the value of Y(k) and an adder to add the complemented value of Y(k) with the set-point value (SP). Finally, a latch is used to store the result of subtraction (ERR(k)(16-bit), 8-bit for integer value and 8-bit for partial).

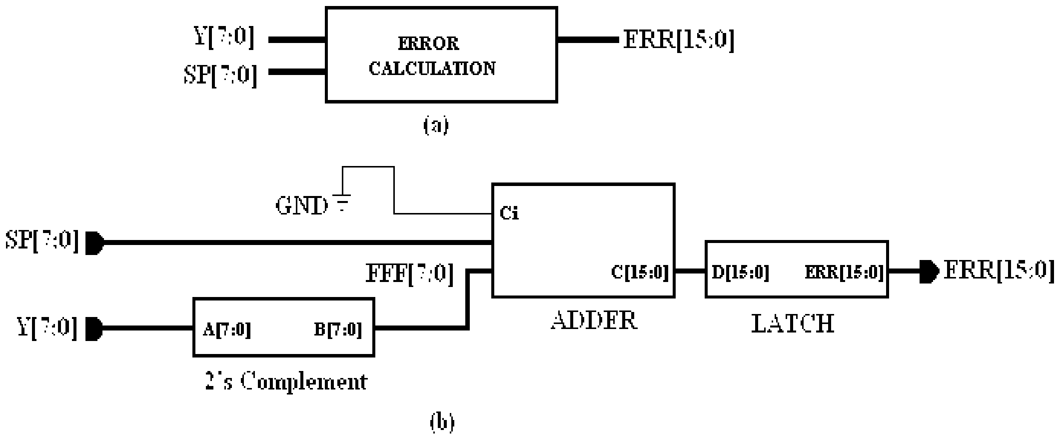


Figure (4) Error calculation part: a) General block, b) detailed block

3-2-2 P Controller

The difference equation of the P controller is:

$$\mathbf{POUT(K) = KP.ERR(K)} \quad \text{... (2)}$$

This part is implemented with the digital circuit that is shown in **Fig.(5)**. The function of the multiplier block is to multiply the error (ERR) with KP. Also, the 2's complement block is an conditional converter that converts the negative number only into a 2's complement value. Finally, the latch block is used to hold the value of P controller for the Adder stage.

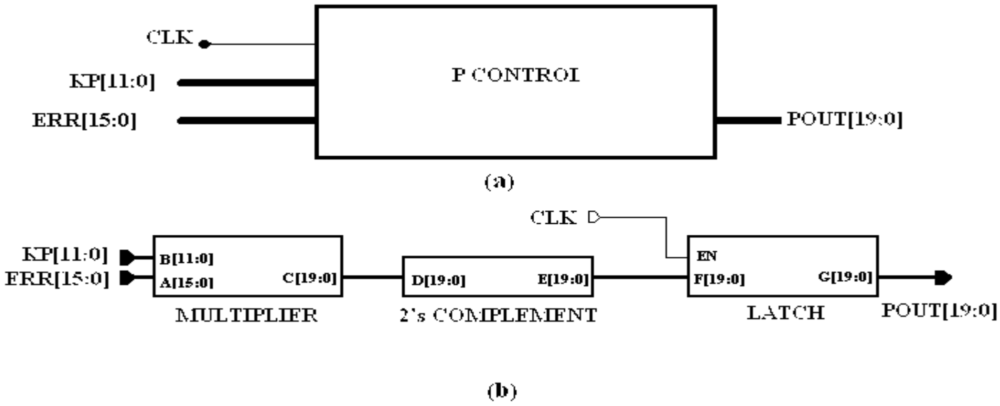


Figure (5) P control part: a) General block, b) detailed block

3-2-3 I Controller

The difference equation of I controller using backward shift approximation is:

$$\mathbf{IOUT(k) = IOUT(k-1) + KI.ERR(k).TS} \quad \text{………………………………... (3)}$$

where: IOUT(k) is the output of the I controller at kth sampling time.

The digital circuit of the I control is shown in **Fig.(6)**. The multiplier block multiplies the three values of TS, KI and ERR. The result is converted to a 2's complement value using 2's complement block. Then the result is added with the past results of integral terms using an adder. Finally, the value is stored in a latch to be used with the final adder.

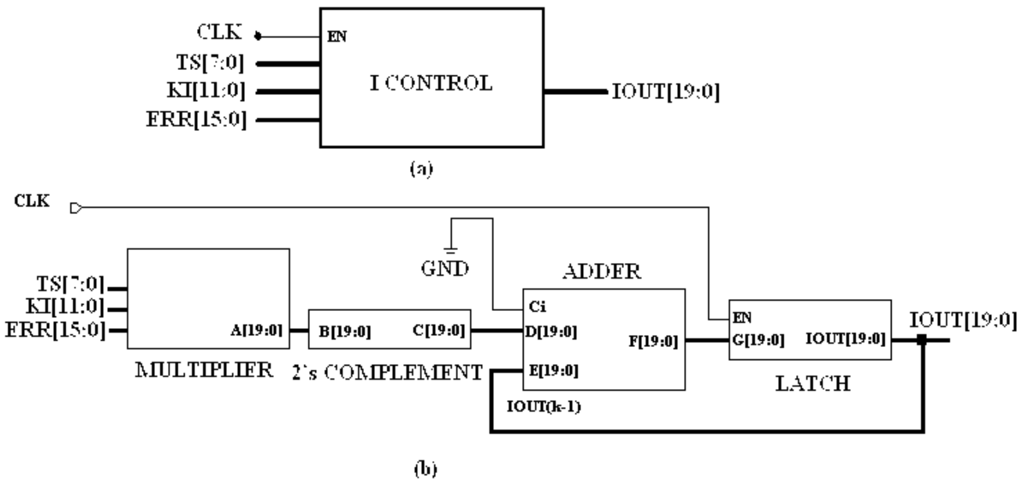


Figure (6) I control part: a) General block, b) detailed block

3-2-4 D Controller

The digital D controller part is implemented using backward shift approximation. The difference equation is:

$$DOUT(K) = \frac{KD}{TS}(ERR(k) - ERR(k-1)) \quad \text{..... (4)}$$

Figure (7) shows the block circuit of D control. It consists in the first stage a divider to divide the derivative constant (KD) on the sampling time (TS), while the other block is used to subtract the previous value of past error (ERR(k-1)) which is stored in an internal latch that enabled by the CLK in this block with the actual error (ERR(k)). The results of the multiplier and subtractor are multiplied in the second stage multiplier to produce the D control output value. This value is stored in a latch in order to be added to the final adder unit.

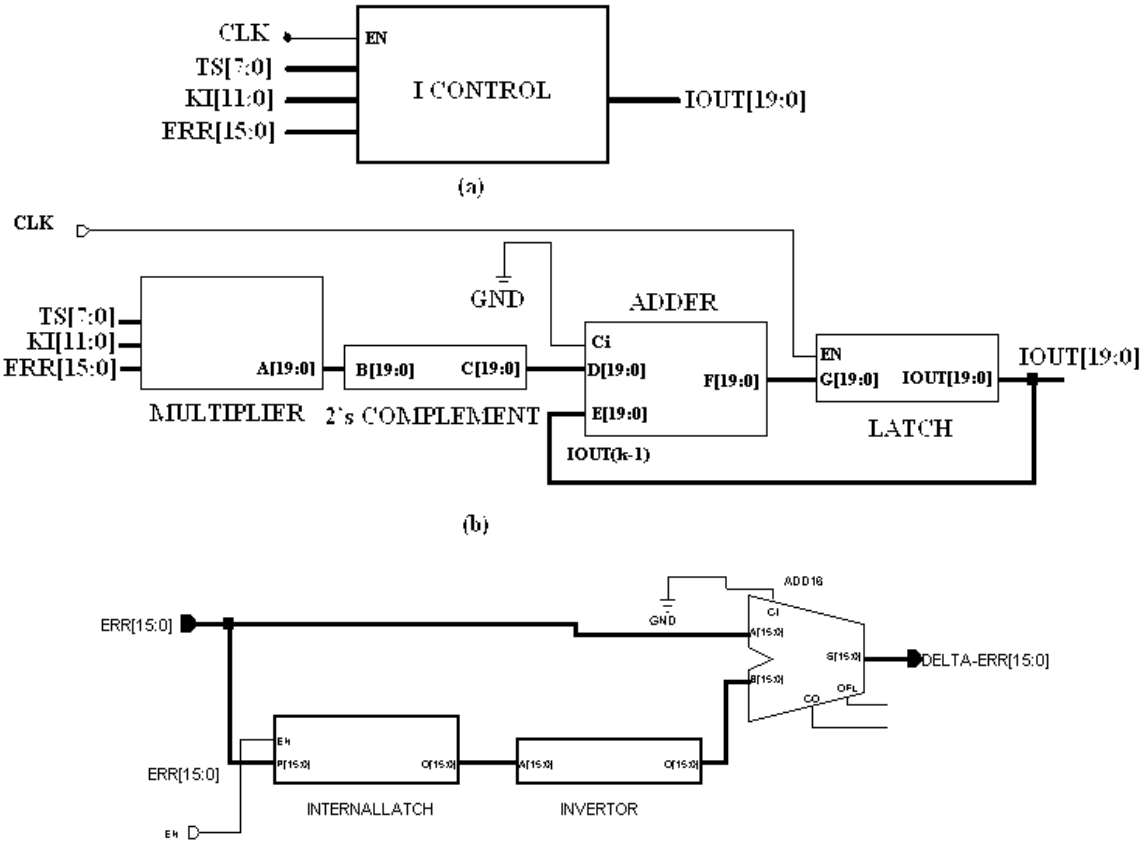


Figure (7) D control part

a) General block,

b) detailed block,

c) details of subtractor

3-3 Final Adder Unit

The final part is the final adder unit that adds all the outputs of the controller parts according to the select control signals, (DC and IC), to select the controller type as (P, PI, PD or PID). The final adder unit is shown in **Fig.(8)**.

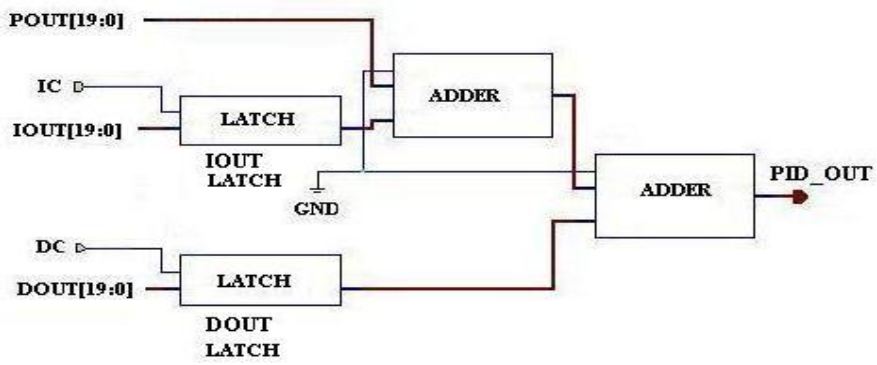


Figure (8) Final adder unit

4. Practical Results

In order to test the system a 1st order discrete linear model is connected with the controller. The difference equation of the model is:

$$\mathbf{Y(k) = 0.5Y(k-1) + 0.5U(k)} \quad \ldots\ldots\ldots\ldots \quad \textbf{(5)}$$

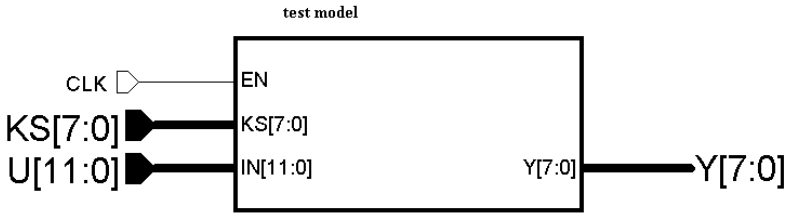
where:

Y(k) : represents the output of the model at kth sampling time.

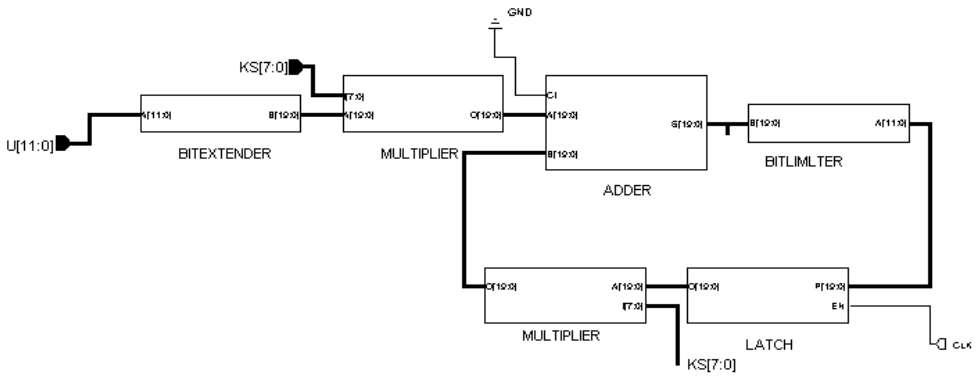
U(k) : represents the output of the controller at kth sampling time, and

KS: represent the constant multiplying value (0.5 in this case).

This tested model is represented by two multipliers and a group of adders, as shown in **Fig.(9)**. However, in order to simulate the controller, U(k) is connected to the PID_OUT of the controller while Y(k) is connected to the input of error block of the controller.



(a)



(b)

Figure (9) a) The test model, b) detailed test model

The FPGA results of the closed loop system are obtained using the simulator of the Xilinx software package (Foundation 3.1i). At the same time, the closed loop system was simulated using MATLAB, SIMULINK program (Ver. 7). Where (the difference between the MATLAB results and FPGA software results is that the first depend on the software calculation according to the theory algorithms, while the other depend on the practical implementation of a digital circuit). Applying a set-point value of (SP=5) for all cases, the simulated controller was tuned using trail and error to get a minimum over shoot and minimum settling time with minimum steady state error. The sampling time was chosen to be 500 msec. As a result, the best values of controller constants were: KP is 0.125, KI is 0.5 and KD is 0.125. The same tuning values with SP=5 were used with the FPGA controller. Different tests were made to check the FPGA controller. The results were shown in **Fig.(10)**.

Figure (10 a) and **Fig.(10 b)** shows the response of the closed loop system with P controller for MATLAB and FPGA systems respectively. While **Fig.(10 c)** and **Fig.(10 d)** shows the response of PI controller for the MATLAB and FPGA systems respectively. Moreover, **Figure (10 e)** and **Fig.(10 f)** shows the response of the closed loop system with PD controller for MATLAB and FPGA systems respectively. Finally, **Figure (10)** shows the response of the closed loop system with PID controller for MATLAB and FPGA systems respectively. It can be concluded from the simulation and practical results that the responses of closed loop system using P and PI and PID controllers were approximately the same while there is a little difference in the closed loop response of PD controller. This is due to the approximation of the design PID controller to the nearest rounding integer.

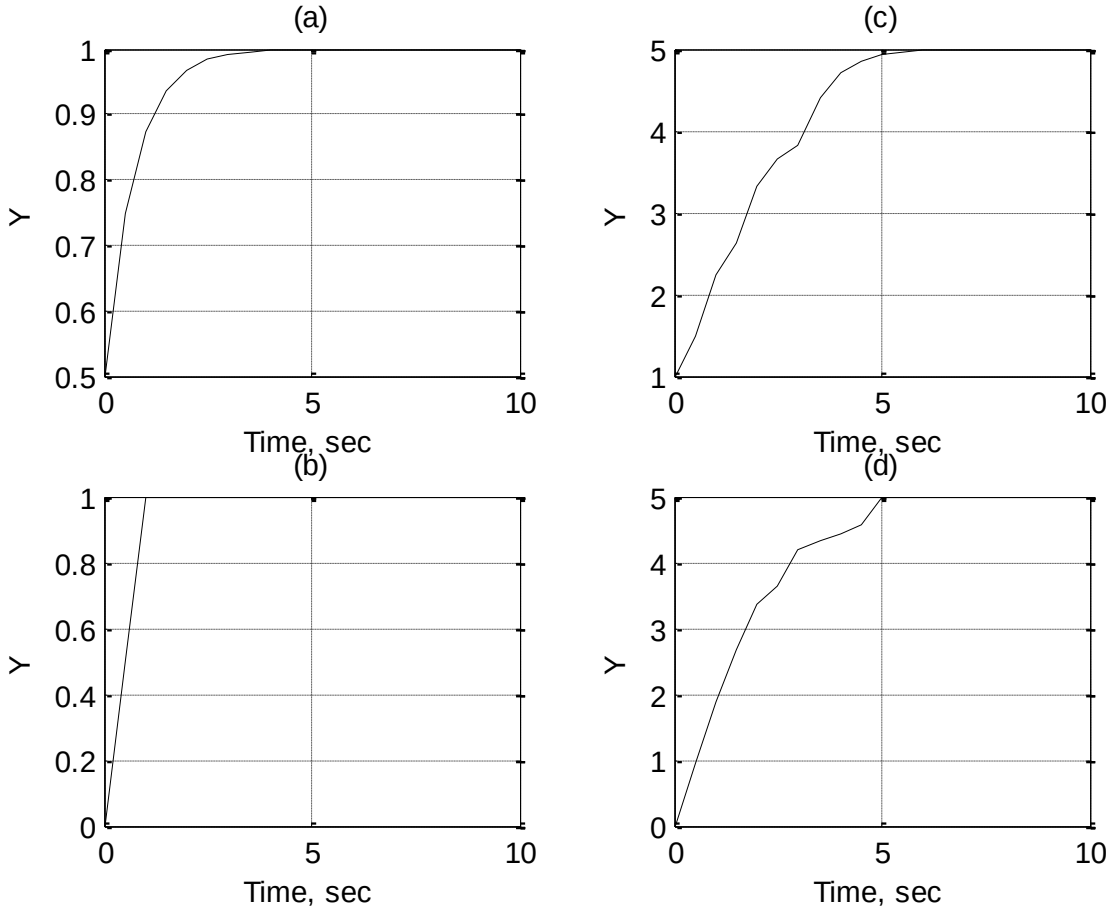


Figure (10) The responses of the closed loop system using MATLAB and FPGA

(a) Response of closed loop system with P controller using MATLAB

(b) Response of closed loop system with P controller using FPGA

(c) Response of closed loop system with PI controller using MATLAB

(d) Response of closed loop system with PI controller using FPGA

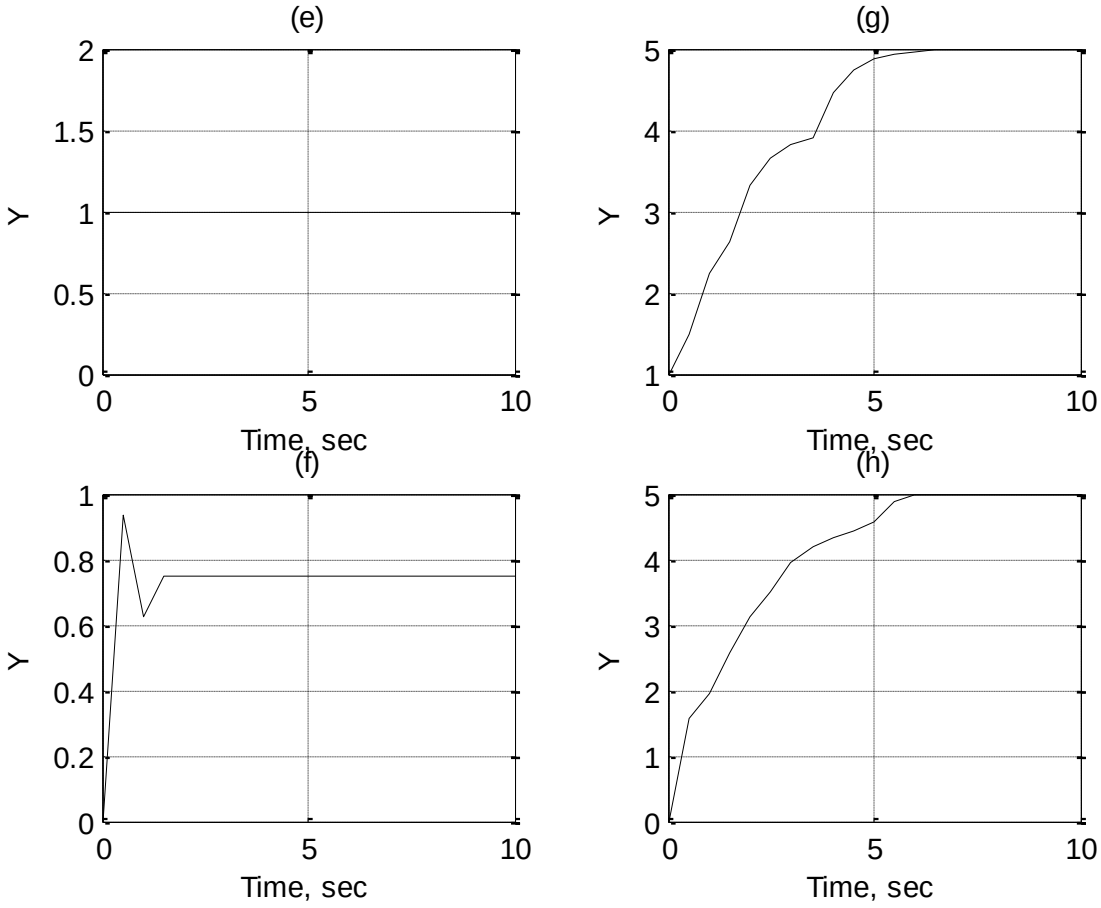


Figure (10) The responses of the closed loop system using MATLAB and FPGA

(e) Response of closed loop system with PD controller using MATLAB

(f) Response of closed loop system with PD controller using FPGA

(g) Response of closed loop system with PID controller using MATLAB

(h) Response of closed loop system with PID controller using FPGA

5. Conclusions

A design of digital controller using FPGA is discussed. The schemes of the controller can be selected to be P, PI, PD or PID type. The design provides a minimization in the hardware and as a result reduction in cost compared with the design of digital computers, since the controller is implemented using an FPGA chip. Practical results were fairly the same as compared with the simulation results produced by the simulation of the system using MATLAB/SIMULINK program.

6. References

1. Kaya, A., and Schelb, T. J., ***"Tuning of PID Control of Different Structures"***, Control Engineering Magazine, Vol. 35, No. 7, July 1988.

2. Gerry, J. P., ***"A Comparison of PID Control Algorithms"***, Control Engineering Magazine, March 1987.

3. Duarte, J. L., Aubry, J. F., and Iung, C., ***"Current and Speed Digital Control of Commutationless DC Drives"***, IEEE Transactions on Industrial Electronics, Vol. 36, No. 4, Nov. 1989.

4. Dewan, S. B., and Dunford, W. G., ***"A Microprocessor Based Controller for a Three-Phase Controlled Rectifier Bridge"***, IEEE Transactions on Industry Applications, Vol. IA-19, No. 1, Jan./Feb. 1983.

5. Sule, R. R., Vasanth, B. J., Krishnan, T., and Kumap, M., ***"Microprocessor-Based Speed Control System for High Accuracy Drive"***, IEEE Transactions on Industrial Electronics, Vol. IE-32, No. 3, Aug 1985.

6. Othman, M. Z., and Al-Shemery, M. J., ***"Different PID Structures Tuned by Genetic Algorithms"***, 1st National Conference on Computer, Communication and Control Systems Engineering, U.O.T., Dec. 2000.

7. Ballagh, J. B., ***"An FPGA-based Run time Reconfigurable 2-D Discrete Wavelet Transform Core"***, Thesis in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering, The Faculty of the Virginia Polytechnic Institute and State University 2001.

8. Brown, S., and Vranesic, Z., ***"Fundamentals of Digital Logic with VHDL Design"***, McGraw-Hill Companies, Inc., 2000.

9. ***"Literature Survey of Present State of FPGA's"***, http://www.microtronix.com, 2000.

10. Saleh, M. H., ***"Design and Implementation of High Repetition Laser Computational System Using FPGA Technology"***, M.Sc. Thesis, Department of Computer Engineering of Military College of Engineering, Baghdad 2002.

11. Yalamanchili, S., ***"Introductory VHDL: From Simulation to Synthesis"***, Prentice-Hall, Inc., 2001.

12. Navabi, Z., ***"VHDL Analysis and Modeling of Digital Systems"***, McGraw-Hill Companies, Inc., 1993.

13. Zwalink, M., ***"Digital System Design with VHDL"***, Pearson Education Limited, 2000.