

# Design of Digital PID Controllers Relying on FPGA-based Techniques

Adriana A. Aguirre, Leonardo D. Muñoz, César A. Martín  
María J. Ramírez, Carlos A. Salazar

*ESPOL Polytechnic University, Escuela Superior Politécnica del Litoral, ESPOL*

*Facultad de Ingeniería en Electricidad y Computación, Campus Gustavo Galindo Km. 30.5 Vía Perimetral  
P.O. Box 09-01-5863 Guayaquil, Ecuador.*

{aaaguirr, leodemun, cmartin, majorami, asalazar}@espol.edu.ec

---

**Abstract:** A major challenge of higher education institutions is to prepare professionals capable of learning by building effective solutions that are able to integrate different disciplines and knowledge areas. With this in mind, academics and researchers from ESPOL university in Ecuador, have designed laboratory experiments for a digital control class that introduces students to a modular design of embedded feedback controllers using Field Programmable Gate Array (FPGA) technologies. The proposed experiment includes the design of direct discrete time PID controllers, for an existing speed control system with three different sampling times, to test and compare their performance. The obtained controllers are implemented using a prototyping strategy that relies on FPGA development boards. The prototype controller is tested using the experimental plant, and the system performance is contrasted with results from simulations under realistic conditions.

*Keywords:* Control education, control design, FPGA, Embedded systems.

---

## 1. INTRODUCTION

In the field of control engineering, the Proportional Integral Derivative controller (PID) strategy is widely adopted because of its simplicity and effectiveness (Jain et al., 2016). For this reason, this topic is still subject of research and testing for both, industrial and academic fields. With the emergence of Industry 4.0 ideas, it is imperative for supervisors to be able to receive data from local and remote systems; this is also supported by the fast development of the Industrial Internet of Things (IIoT) field, that focuses on transfer and control of critical information (Carrasco, 2017). With these ideas in mind, the design of control systems (e.g., PID controllers) implemented over fundamental digital platforms (e.g., FPGA prototyping) is relevant to allow the application of any specific communication and development protocol, without the limitations imposed by commercial brands.

Some of the PID implementation strategies use low cost open source programmable electronic devices (Dobra et al., 2009). Others have studied an implementation using different types of micro controllers, but still there are few works about digital controllers implemented over Field Programmable Gate Array (FPGA) for education purposes.

FPGA boards are included in automation processes due to their advantages like flexible controller programming, and expandable implementation according to particular needs. They also have a higher speed response at a lower price in comparison with other systems; their algorithmic solving

process is performed in parallel with the capability of working with multiple inputs and multiple outputs systems (MIMO) (Aboelaze and Shehata, 2015). Also, some FPGA boards can send data to cloud web servers and this can help to prevent issues before they appear (Hanafi and Karim, 2015).

The purpose of this paper is to develop a set of laboratory experiments, at ESPOL Polytechnic University, for a digital control class where PID controllers are designed to optimize the performance of a speed control system using embedded systems. One of the advantages in the use of embedded systems is that the architecture can be adapted according to particular industrial or user requirements, optimizing resources. Currently, embedded systems are used as solutions in different settings, like security systems or traffic notifications with lower costs than conventional analog controller systems (Castillo and Niebieskikwiat, 2014). Students are able to design and implement a stand alone independent controller for a prototype system.

This paper is structured as follows. Section 2 describes the evolution of controllers in industry. Also, general information about the system to be used in the experiment, is presented. Section 3 presents the design of a digital PID controller and its implementation based on an FPGA. Section 4 presents simulation and implementation results. Section 5 gives a summary of conclusions and describes future work.

## 2. BACKGROUND

Human beings have figured out different ways to improve their production processes. With the first industrial revolution, industries were capable to substitute hand made processes with machines. In this way, the quantity and quality of production increased greatly. Later, the use of other sources of energy was encouraged for serial production. Then the third industrial revolution took place with the development of information technologies for processes automation. Finally the fourth industrial revolution is taking place in many countries; it is spreading rapidly and faster than the previous ones. It is also called the digital revolution and it joins and mixes different types of technologies like Big Data, Cloud Computing, Internet of Things and System Integration, among others.

During its history, and in constant pursuit of excellence, ESPOL has proposed many curricular reforms that guarantee that the contents and the scope of the programs are up to date. The last curricular reform was performed in 2016, and was carried out in the context of the current legislation. Among other important topics, the reform included curricular organization and training in science, analysis and problem solving, that help students to be focused on technology and research as a mean to address important problems in the country.

Although digital control principles have been taught on ESPOL in their graduate programs for 7 years, the continuous presence of digital electronics in daily life makes necessary to teach these concepts to undergraduate students. Due to the aforementioned aspects and industry demands, a digital control systems course is introduced to the electronics and automation engineering undergraduate program. The course includes a practical component; hence, laboratory experiments must be developed.

### 2.1 Speed Control System

The Control Systems Laboratory at ESPOL is a facility that allows students to apply the knowledge acquired in the theoretical control systems course. The laboratory has different prototype plants of common industrial control systems including speed, level and position controls. In addition, students who attend the lab are able to access computers with the MATLAB™ software with the control system toolbox installed.

The prototype plant for speed control is designed to maintain the velocity of a motor through a voltage applied to a variable frequency drive, which has a range of 0 to 5 V. The input voltage produces a frequency range from 0 to 60 Hz, which is converted to an output speed between 0 and 1590 rpm. The system includes an electrically operated valve with an input range from 4 to 20 mA. The position of the valve changes the oil pressure that is pumped through the system. The pump, which is connected to the motor shaft, allows oil circulation from the reservoir, through the system and back to the reservoir. The pump works as a load for the motor. System measurements are gathered in real time, and are initially collected through a data acquisition card for the identification process, then the data acquisition card is replaced with a FPGA board that

will have the digital PID controller implemented. A photo of the prototype is presented in Fig. 1.



Fig. 1. Photo of the Speed Control System.

To design a proper controller, it is necessary to find a mathematical model that represents adequately the dynamic nature of the system. One way to experimentally find a model, is to develop a system identification experiment; this is performed by exciting the system with a basic step input signal, and at the same time measuring the output; finally, input-output data is processed using the system identification toolbox from MATLAB™. This toolbox uses Prediction Error Methods (PEM) (Tran et al., 2018) to find an appropriate process model with sufficient prediction capabilities, and relying on the measured information. A set point of 2.5 V was selected from the characteristic curve of the system, that corresponds to a system output of 715 rpm and a set-point of 12 mA was selected for the electro valve. The obtained open loop transfer function for this set point using system identification is:

$$G_p(s) = \frac{\Omega(s)}{V(s)} = \frac{4251.6}{(s + 3.449)(s + 3.855)} \quad (1)$$

It is important to mention that this function represents the ratio of the output speed of the motor  $\Omega$  to the input voltage of the system  $V$  in the Laplace domain considering its initial conditions and equilibrium point; this means that the input used in the transfer function corresponds to the real input minus the input set point (2.5 V) and the output obtained from the transfer function plus the output set point (715 rpm) is the real output of the system. This is commonly known as incremental model and is used in nonlinear systems representation (Guru, 2015).

### 2.2 FPGAs

The DE0-Nano is a board (see Figure 2) that includes FPGA technology with other components such as Analog-Digital Converter (ADC), synchronous dynamic random-access memory (SDRAM), accelerometer, 2 40-pin Input/Output (GPIO), flash memory and I2C EEPROM.

The DE0-Nano FPGA model is Cyclone IV E family EP4CE22F17C6; it receives a 50 MHz clock signal from the quartz crystal on the board (Terasic, 2016).

Quartus Prime™ is a design software from Intel™ that allows users to develop and implement digital systems for FPGAs. Quartus™ uses ModelSim™ interface which is another software used for hardware description languages (HDL) simulation such as VHDL, Verilog and System C (Intel, 2015). These softwares are available in the Digital Systems Laboratory in ESPOL University. There are other tools based on FPGA, such as VIVADO™ from Xilinx™, but the advantage is that the Quartus™ environment is more educational, and Intel™ FPGA chips are cheaper and have similar performances than Xilinx.

The Digital Systems Laboratory has DE0-Nano boards that are flexible, of high performance and low cost. By using MATLAB™ Digital Signal Processing (DSP) Builder tool of MathWorks, students can design and program DE0-Nano boards within MATLAB™ environment.

The use of FPGAs in ESPOL allows students, in particular from electronic courses, to work in the development of digital systems that demand parallel computational processing. For this reason, the laboratory practice is intended to involve these students in the use of FPGA resources for the design and development of discrete controllers in the Control Systems laboratory.

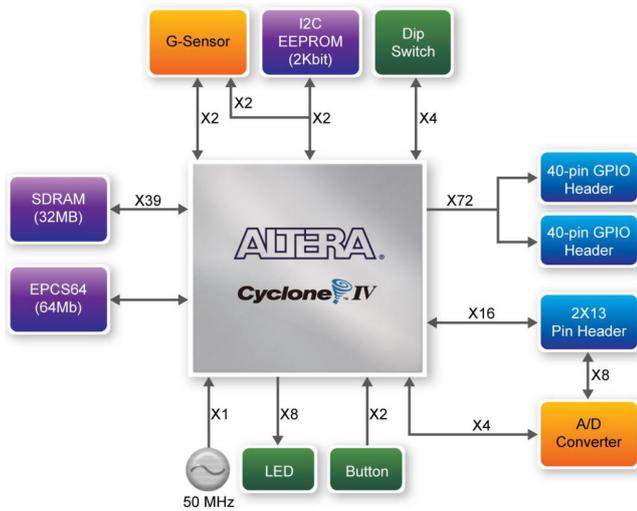


Fig. 2. Block Diagram of DE0-Nano Board (extracted from DE0-Nano User Manual, version 2.0, Terasic™)

### 3. DESIGN AND IMPLEMENTATION OF A DIGITAL PID CONTROLLER

This section describes the proposed laboratory experiment to be applied to the speed control prototype plant described in the previous section. Theoretically PID analog controllers should show a better performance than their digital versions. Most of the time in real processes, factors like computer resources, length of production batches, cost and simplicity, lead control engineers to choose digital PID controllers. Some of the advantages of this type of controllers were mentioned in Section 1.

A digital PID controller will be directly designed in discrete domain using root locus methods. The resulting controller has the form of (2).

$$G_c(z) = K_{pD} + \frac{K_{iD}}{1 - z^{-1}} + K_{dD}(1 - z^{-1}) \quad (2)$$

#### 3.1 Design

The controller design requirements for the speed control prototype plant are:

- Steady State Error for a step input: 0
- Overshoot: 0 %
- Settling time: 4 s.

PID controllers are usually designed to have zero steady state error for a step input, in this way the system's output will try to follow the system's input. The system's overshoot is required to be zero percent so that the system does not have speed peaks but a smooth response. Finally the settling time was chosen to be of five seconds so that students are able to difference the response of the system in form of speed change when a step input is applied.

The digital controllers will be designed and tested for 3 different sampling times:  $T = 0.0125$  s,  $T = 0.125$  s and  $T = 0.75$  s. These values were chosen according to the Nyquist–Shannon sampling theorem. If a sufficient sample rate is used, the original signal can be reconstructed from the sampled signal. The theorem states that the sufficient sample rate is equal to  $2B$  where  $B$  is the highest frequency of the signal to be sampled. The three sampling times represent oversampling, sufficient sampling and under-sampling respectively.

A discrete time representation of the system transfer function is obtained with a sampling time  $T$ , and including the effect of a Zero Order Hold (ZOH) block, as is described in (Fadali, 2013). The ZOH block holds the signal value between consecutive sampling instants. The  $z$ -domain discrete time representation depends directly on the value of the sampling time  $T$ ; thus, different sampling times result on different discrete transfer functions and different digital controller realizations (Ogata, 1995). To represent more accurately the real-time performance of the system, the discrete transfer function must incorporate a delay caused by internal components in the data acquisition card. This delay is considered to be equal to one sampling time.

For  $T = 0.0125$  s,  $T = 0.125$  s, and  $T = 0.75$  s the computed open loop discrete transfer functions  $G(z)$  are respectively:

$$G_1(z) = \frac{0.32222(z + 0.97)}{z(z - 0.9578)(z - 0.953)} \quad (3)$$

$$G_2(z) = \frac{24.64773(z + 0.7374)}{z(z - 0.6498)(z - 0.6176)} \quad (4)$$

$$G_3(z) = \frac{242.03(z + 0.1539)}{z(z - 0.07526)(z - 0.05551)} \quad (5)$$

Parameters for discrete time PID controllers (i.e.,  $K_{pD}$ ,  $K_{iD}$ , and  $K_{dD}$ ) are found using the control system designer application from the Control System Toolbox in MATLAB™, and fundamental principles described in (Fadali, 2013; Ogata, 1995). The closed loop, controller and

open loop transfer functions have the form of Equations 6, 7 and 8 respectively.

The control strategy used was to locate the zeros of the controller in the same position as the system's poles. An integrator was added to the controller so that the zero steady state error is achieved. Then the controller gain is adjusted so that the other design requirements are met.

$$T(z) = \frac{C(z)GH(z)}{1 + C(z)GH(z)} \quad (6)$$

$$C(z) = \frac{K(z - z_1)(z - z_2)}{z(z - 1)} \quad (7)$$

$$GH(z) = \frac{K_g(z - z_g)}{z(z - p_1)(z - p_2)} \quad (8)$$

, where  $z_1$  is equal to  $p_1$  and  $z_2$  is equal to  $p_2$ . Then, the characteristic equation of the system is defined by Equation 10.

$$1 + C(z)GH(z) = 0 \quad (9)$$

$$z^2(z - 1) + K_gK(z - z_g) = 0 \quad (10)$$

$$z^3 - z^2 + K_1z - K_1z_g = 0 \quad (11)$$

One of the required poles is  $z = 0.988$ . Substituting this value in the previous equation:

$$(0.988)K_1 - (K_1z_g + 0.0117) = 0 \quad (12)$$

$$(0.988 - z_g)K_1 = 0.0117 \quad (13)$$

Substituting each open loop transfer function zero  $z_g$  in Equation 13,  $K_1$  can be calculated. Then, the corresponding obtained digital PID controllers for  $T = 0.0125$  s,  $T = 0.125$  s and  $T = 0.75$  s are respectively:

$$C_1(z) = \frac{0.0192z^2 - 0.0366z + 0.0175}{z^2 - z} \quad (14)$$

$$C_2(z) = \frac{0.002363z^2 - 0.002995z + 0.0009482}{z^2 - z} \quad (15)$$

$$C_3(z) = \frac{0.000799z^2 - 0.0001045z + 3.337e - 06}{z^2 - z} \quad (16)$$

For  $T = 0.75$  s, the settling time requirement could not be accomplished. The controller for this sampling time was adjusted for a settling time of 8 seconds.

### 3.2 Implementation

The tool used to model the digital PID controller was DSP Builder. This tool is commonly used to model digital systems from MATLAB<sup>TM</sup> environment. The system receives two inputs from one analog-digital converter, that is included in the DE0-Nano board. One input is the reference and the other is the output bus signal of the plant. The system has one output that at the same time corresponds to the input of the prototype plant. The ADC block was compiled in Quartus Prime<sup>TM</sup>, then it was imported in the DSP project using the HDL block from the toolbox DSB Builder for Intel FPGAs - Standard Blockset. The ADC block has up to 8 analog channels; the other signals are used

to communicate via SPI protocol with the ADC128S022 chip present on the board. Figure 3 shows the Analog-Digital Converter Hardware Description Language block that was used in the model. The rest of the blocks were obtained from DSP Builder toolbox such as Integrator, Differentiator and Gain.

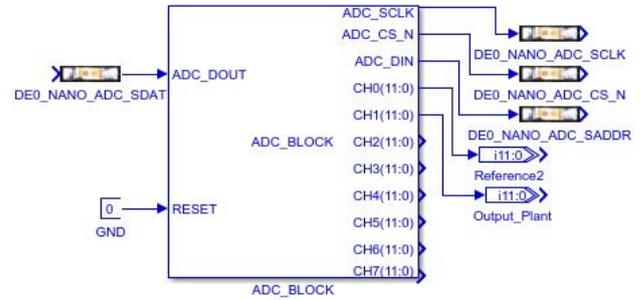


Fig. 3. ADC Block.

## 4. RESULTS

The speed control prototype plant was tested with three different controllers. Each controller was designed to meet the same design requirements for the same plant but with different sampling times. The first controller was designed with  $T = 0.0125$  s, the second controller was designed with  $T = 0.125$  s and the third controller was designed with  $T = 0.75$  s. The PID Control rule follows 2.

### 4.1 Simulation

Simulink and ModelSim were used to simulate the system's response with each controller; Figure 4 shows the digital PID Controller structure used in Simulink for the simulation. This environment is used to set the PID parameters, sampling time and transfer function of the prototype plant to be simulated.

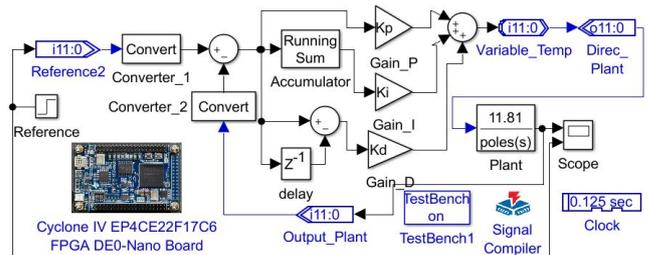


Fig. 4. PID Controller Block Diagram.

Simulation results were observed in Matlab<sup>TM</sup> environment using the software's commands. Figures 5, 6 and 7 show the system response for controllers  $C_1$ ,  $C_2$  and  $C_3$  respectively.

### 4.2 Implementation

The control signal applied to the motor driver is a PWM signal, this block receives a 12-bit signal and it has as output the PWM signal at a frequency of 100 kHz; the voltage levels of the output are in the range of 0 to 5 V.

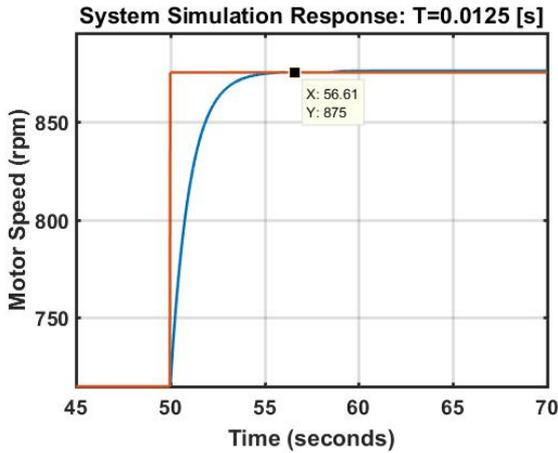


Fig. 5. Simulation of system response for controller  $C_1$  ( $T=0.0125$ ).

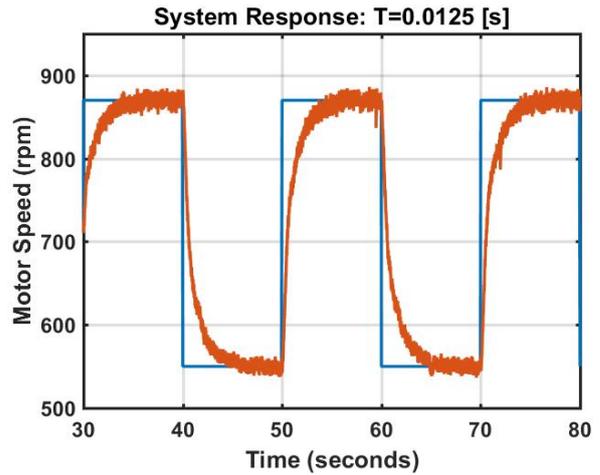


Fig. 8. System Response for controller  $C_1$  ( $T=0.0125$ ).

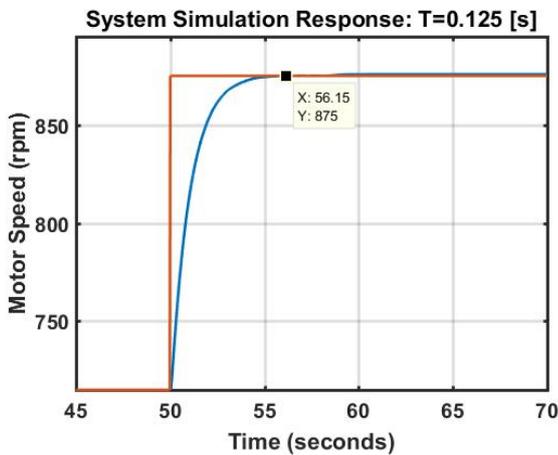


Fig. 6. Simulation of system response for controller  $C_2$  ( $T=0.125$ ).

Figure 9 represents the system response when controller  $C_2$  was implemented. The steady state error, settling time and overshoot design requirements were also met.

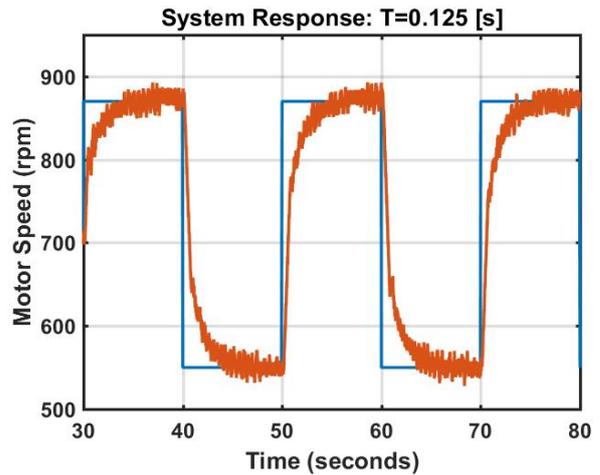


Fig. 9. System Response for controller  $C_2$  ( $T=0.125$ ).

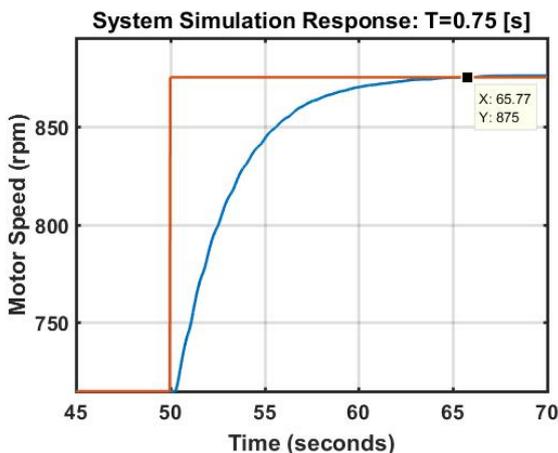


Fig. 7. Simulation of system response for controller  $C_3$  ( $T=0.75$ ).

Figure 10 represents the system response when controller  $C_3$  was implemented. The steady state error, and overshoot design requirements were also met but the settling time requirement was not accomplished following the same design strategy as the other controllers. This is more noticeable on Figure 11 where two out of the three outputs of the system are similar.

Figure 8 represents the system response when controller  $C_1$  was implemented. The steady state error, settling time and overshoot design requirements were met.

Figure 11 includes in the same graphic the system response with each controller in order to compare their performances. The input signal period was set to 20 seconds considering that the settling time for  $T = 0.75$  s is 8 seconds. Figure 11 also shows that the system responses for  $T = 0.0125$  s and  $T = 0.75$  s are very similar; the blue line that represents the system response for  $T = 0.0125$  s can not be appreciated because the system response for  $T = 0.0125$  s is superimposed.

The results obtained from the implementation are summarized in Tables 1, 2 and 3.

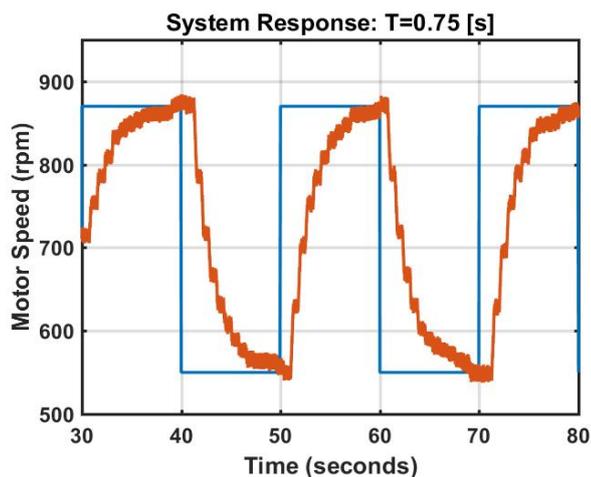


Fig. 10. System Response for controller  $C_3$  ( $T=0.75$ ).

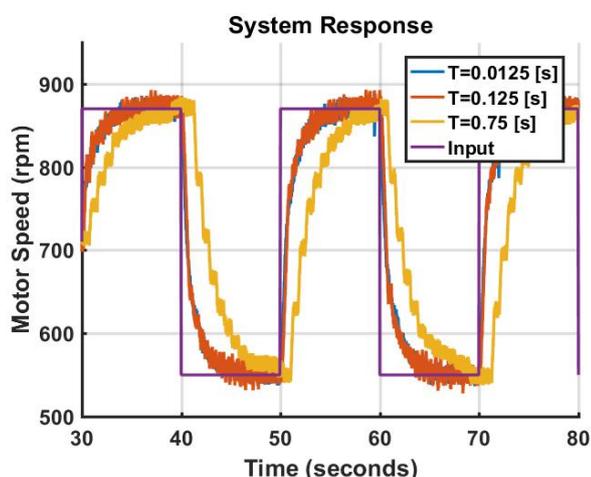


Fig. 11. System Responses for all controllers .

Table 1. Results:  $T=0.0125$

|          | Steady State error | % Overshoot | Settling time |
|----------|--------------------|-------------|---------------|
| Designed | 0                  | 0           | 4             |
| Real     | 0                  | 0           | 3.95          |

Table 2. Results:  $T=0.125$

|          | Steady State error | % Overshoot | Settling time |
|----------|--------------------|-------------|---------------|
| Designed | 0                  | 0           | 4             |
| Real     | 0                  | 0           | 4.02          |

Table 3. Results:  $T=0.75$

|          | Steady State error | % Overshoot | Settling time |
|----------|--------------------|-------------|---------------|
| Designed | 0                  | 0           | 8             |
| Real     | 0                  | 0           | 7.98          |

## 5. CONCLUSIONS AND FUTURE WORK

In this work, digital PID controller is implemented in an FPGA board in a simple and flexible way. Students are able to observe the actual digital implementation of the controller, and study the impact of the sampling time selection. From the academic point of view, learning horizons were expanded since two different study fields were used together for this practical experiment, control systems and digital electronics including embedded systems as well. As was demonstrated in the previous section, a

greater sampling time reduces the overall performance of the system, and different control strategies are needed in order to achieve the same performance if a controller with a smaller sampling time was to be used. As a result of this work, a guide for a lab practice for the digital control course was created.

As future work, communication between FPGA target and a remote computer is recommended. In this way students can test their controllers, during established hours, without being physically on the Control Systems Laboratory; a continuous monitoring process can also be established in order to prevent future errors and to detect early failures.

## REFERENCES

- Aboelaze, M. and Shehata, M.G. (2015). Implementation of multiple PID controllers on FPGA. *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 446-449.
- Carrasco, M. (2017). 4ta. revolución industrial. *FOCUS*, 22-27.
- Castillo, V. and Niebieskikwiat, J. (2014). Los sistemas embebidos y su importancia en la actualidad. *J. Brit. Med. Assoc.* URL <http://licuadodeletras5.wordpress.com/los-sistemas-embebidos-y-su-importancia-en-la-actualidad/>.
- Dobra, P., Dumitrache, D., Tomesc, L., Duma, R., and Trusca, M. (2009). Low-cost embedded solution for PID controllers of DC motors. *2009 17th Mediterranean Conference on Control and Automation*, 1178-1183.
- Fadali, M.S. (2013). *Digital Control Engineering Analysis and Design*. Elsevier Science, Burlington, 2nd edition.
- Guru, C. (2015). Design and tuning recipe must consider nonlinear process behavior. URL <http://controlguru.com/design-and-tuning-recipe-must-consider-nonlinear-process-behavior/>.
- Hanafi, A. and Karim, M. (2015). Embedded web server for real-time remote control and monitoring of an FPGA-based on-board computer system. *2015 Intelligent Systems and Computer Vision (ISCV)*, 1-6.
- Intel (2015). Altera's new quartus prime design software extends leadership in design performance and productivity.
- Jain, R.V., Aware, M.V., and Junghare, A.S. (2016). Implementation of a PID control PWM module on Altera DE0 kit using FPGA. *2016 IEEE First International Conference on Control, Measurement and Instrumentation*, 341-345.
- Ogata, K. (1995). *Discrete-time control systems*. Prentice Hall, Englewood Cliffs, N.J., 2nd edition.
- Terasic (2016). *DE0 Nano User Manual*, version 2.0 edition.
- Tran, L.T.T., Nordholm, S.E., Schepker, H., Dam, H.H., and Doclo, S. (2018). Two-microphone hearing aids using prediction error method for adaptive feedback control. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(5), 909-923.