

Teaching Tool for Digital Control and Signal Processing Generating Automatic Code for FPGA's

Edilberto Carlos Vivas Gonzalez, Diego Mauricio Rivera and Edwar Jacinto

Universidad Libre, Universidad Pedagogica Nacional and Universidad Distrital Bogota D.C., Bogota, Colombia
edilbertoc.vivasg@unilibrebog.edu.co, dmriviera@pedagogica.edu.co, ejacintog@udistrital.edu.co

Abstract

Objectives: The educational implementation of code generation software for FPGA device is presented. The objective is to allow the students to develop implementations without knowing the structure of the programmable logics device, or any hardware description language. **Methods:** A software tool was developed for this purpose. In the user's interface of the tool, the transference function of the controller or digital filter must be entered, and then the code VHDL will be automatically generated in a standard format, compatible with any programmable logics device. **Findings:** It was found that the possibility that students might have by implementing a controller, or a higher-order system without making a complex code makes that the learning process speeds up, and that the student faces specific problems related to digital control and signals processing, leaving aside the restrictions given by the difficulties in the use of the hardware's description language, as well as the required knowledge of the devices structure. The use of the tool allows that students develop their laboratory practices generating digital dynamic systems in a high level, which allows avoiding potential errors in their implementations. Besides the possibilities in the industrial level, the tool is capable of being used by students of Engineering without knowledge in hardware programming, which study topics related to control, but not specialized in topics related to implementation. **Application:** Results linked to learning in some engineering classes with and without the use of the tool are shown at the end of this paper.

Keywords: Control Systems, Education, Engineering Teaching, FPGA Code Generation, Signal Processing

1. Introduction

A topic of permanent interest for engineering teachers is related to the need of creating lab activities that make possible to connect the theory with the industrial applications. There are several works that allow to support the control classes with developments to generate controllers for specific problems^{1,2}. It is common to see that laboratory practices for digital control have plants to be controlled through an interface to a computer^{3,4}, connecting a data acquisition card to read the set-point signals

and generating the control actions on their actuators, but this kind of solutions work in plants with slow dynamics, or when the equation in filter differences or digital control is of low-order, since the speed communication between the PC and the DAQ (Data Acquisition) is limited, affecting considerably the systems performance. There are educational options to ease control practices by the usage of pre-programmed hardware^{5,6}.

The typical alternative in practices is making the theory design of the system's controller, and then program a digital device of the embedded type, which involves a

*Author for correspondence

great ability by the designer so the generated code reaches the necessary sample rate, the appropriate numeric format, decide the records' size, truncation, rounding, besides avoiding the passing of available resources of the programmable device. During the evaluation process of control concepts, it has been noticed that a big number of students have trouble in the implementation stage, and not necessarily with concepts related to digital control.

This work presents the application of the described tool⁷ which allows the implementation of multivariable digital controllers and filters in a great variety of applications in several fields⁸⁻¹³ due to its capacity to generate automatically VHDL code. An important feature of the tool is that the code generated is easily scalable and implementable in any FPGA device. In addition, it is compatible with the Black Box of the Toolbox System generator or Simulink. The interface shown by this work is one of high level that only requires entering information such as the state equation or the transfer function of the digital control or filter, which might be of multiple inputs and outputs, the size of the word of input and output, along with the number of bits of the coefficients' records. With this information, the program makes automatically all

the calculation needed for a successful implementation. The resulting VHDL code is totally parallel, that means that the code generated allows the execution of the equation calculation in differences of a clock cycle, making possible simulations and implementations of high speed without any trouble. The tool was used in the digital control and digital signals processing classes of Universidad Distrital Francisco Jose de Caldas, Universidad Libre, and Universidad Pedagogica Nacional for three academic terms. The students' performance was compared to previous terms without the usage of this application. The paper is organized as follows. In Section 2 Concept headings, Section 3 shows the test of the tool. Section 4 shows some results and discussion. Finally, conclusions are presented.

2. Concept Headings

2.1 Define

The general form for digital systems defines through the equation (1) When the constants a_k y b_k are the coefficients:

$$y(n) = b_0x(n) + b_1x(n-1) + \dots + b_Mx(n-M) - a_1x(n-1) - a_2x(n-2) - \dots - a_Nx(n-N) \quad (1)$$

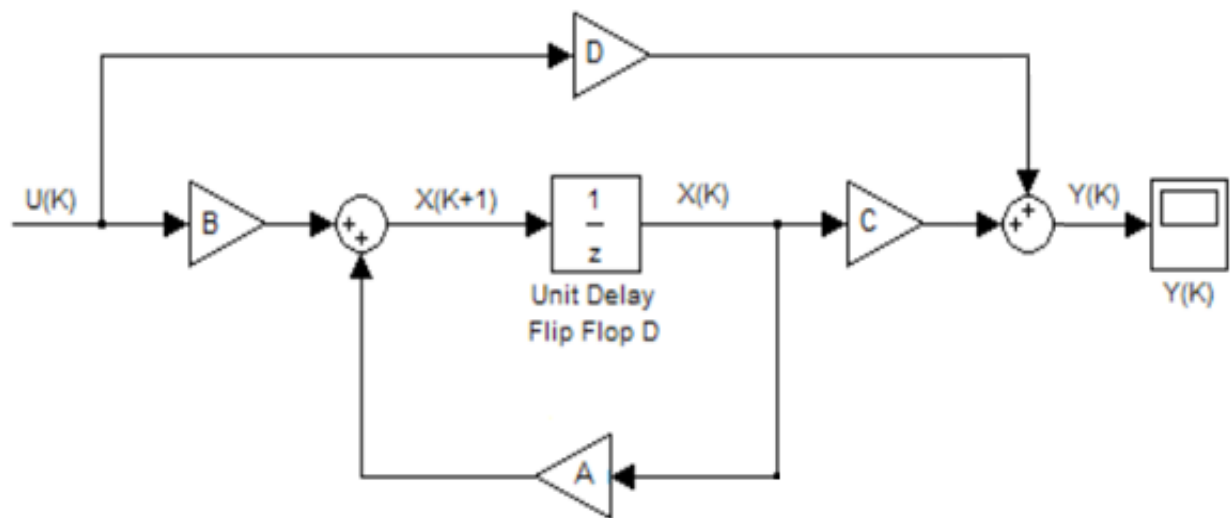


Figure 1. Implementation of IIR filters by state equation using Simulink.

The implementation of digital controls and filters can be made in different ways; for example, the Direct Form I and Direct Form II. However, a better performance is obtained by carrying the transfer function to a state space representation, because this allows minimizing the errors by numeric representation of the coefficients when making balanced realizations. Nevertheless, the coefficients of the realization are generally not zero, which implies an increase in the number of operations, and therefore, the use of resources in the digital device.

A system has infinite representations by choosing a different set of state variables. The implementation of this kind of implementation is shown in the Figure 1. The transformation allows that any kind of controller or digital filter, whether in transposed form I or II, or simply transference function, it can be taken to state variables. That is why the student can enter the system equations in any of these ways, or just by writing the coefficients of the state variable matrix directly.

2.2 Balanced Realizations

The student must understand that digital devices, in this case a FPGA, do not have hardware to make operations in floating point. Due to it, all the coefficients must have a correct representation in fixed point, and they must be in close orders of magnitude to decrease numerical error.

Balanced forms are useful because of their low sensitivity in implementations in finite precision devices. Calculating the balanced form is as follows: a system K

$= [A; B; C; D]$; W_c the Controllability Gramian and W_o the Observability Gramian, also if A is stable, there is a solution to equation (2) and equation (3) if and only if K is controllable and observable.

$$AW_c + W_c A' = -BB' \quad (2)$$

$$AW_o + W_o A' = -CC' \quad (3)$$

And there is a transformation T such that $W_o = W_c$ the observability and Controllability Gramian are equal and is given by equations 4 and 5.

$$W_c = \sum_{k=0}^{\infty} U'(R')^{-1} W_c R^{-1} U \sum_{k=0}^{\infty} \quad (4)$$

$$W_o = \sum_{k=0}^{\infty} U' R W_o R' U \sum_{k=0}^{\infty} \quad (5)$$

2.3 General Block Diagram

For the implementation of the controller or the filter, it is necessary to calculate the number of bits required to

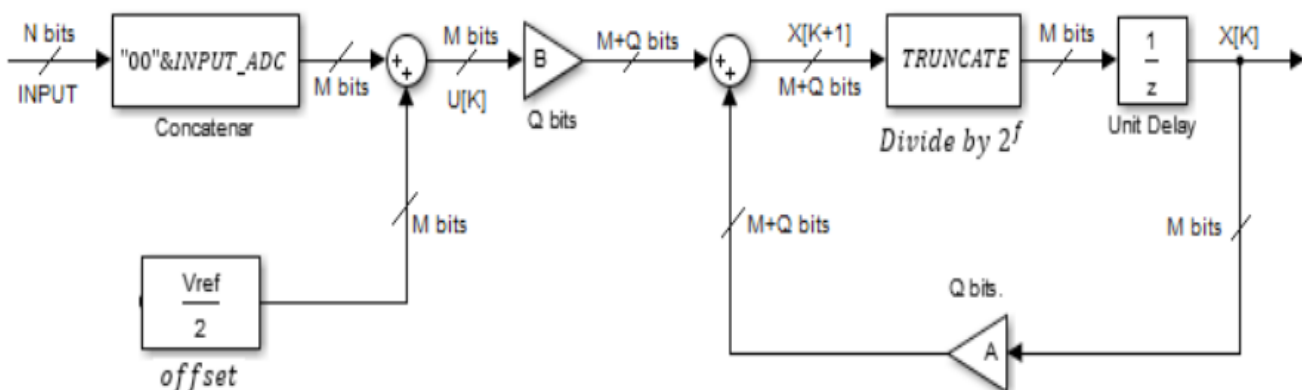


Figure 2. Input stage of the IIR filter⁷.

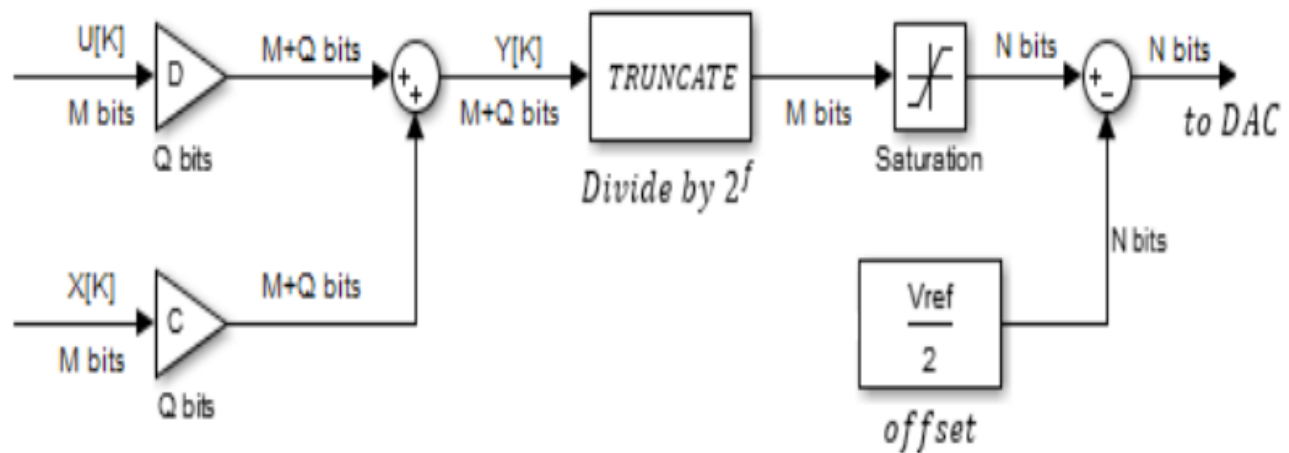


Figure 3. IIR filter output stage⁷.

represents the matrix coefficients of the state equation, starting with the bit number converters both analog to digital (system input) (Figure 2). As well as digital to analog (system output) (Figure 3).

Taking the infinite H standard matrices of the state equation of the system there is an automatic calculation of the size of the records to represent them using the equation 6 and 7.

$$\gamma_A = \|A\|_{\infty} \quad \gamma_B = \|B\|_{\infty} \quad \gamma_C = \|C\|_{\infty} \quad \gamma_D = \|D\|_{\infty} \quad (6)$$

$$size_{x,y} = \log_2 \left(\left(\gamma_C (\gamma_A (\gamma_A + \gamma_B) + \gamma_B) + \gamma_D \right) (2^N) (n + n_u) \right) \quad (7)$$

As there is a representation in fixed point it is necessary to make truncations and rounding so that the result of $Y(K)$ of each interaction accomplishes the appropriate size of word. The structure used makes the calculation of the difference equation in a one clock cycle, but it needs a hardware multiplier for each element of the state matrix,

which enlarges the speed answer of the implemented system, but it means a big usage of resources according to the order of the digital system.

3. Tests of the Tool

In Figure 4 the simulation of a class activity is shown. There students see, as an example: design, implementation, and simulation of a filter under order 4, which was design using the different executions. All of this for the student to understand the difference of design, as well as the verification of the advantages that involve using state variables and balanced realizations.

Unit step response of the deployed filter in direct form I, form II and in direct state equation with a balanced realization was simulated. The filter coefficients are represented in fixed-point numbers with a word-length of 8-bits, 6 fractional bits, 1 integer bit and 1 sign bit. Figure 5 shows that the direct form I and direct form II implementations have a gain error of 37.5%, due to numerical error committed, while the state equation implementation has a gain error of 2.5%.

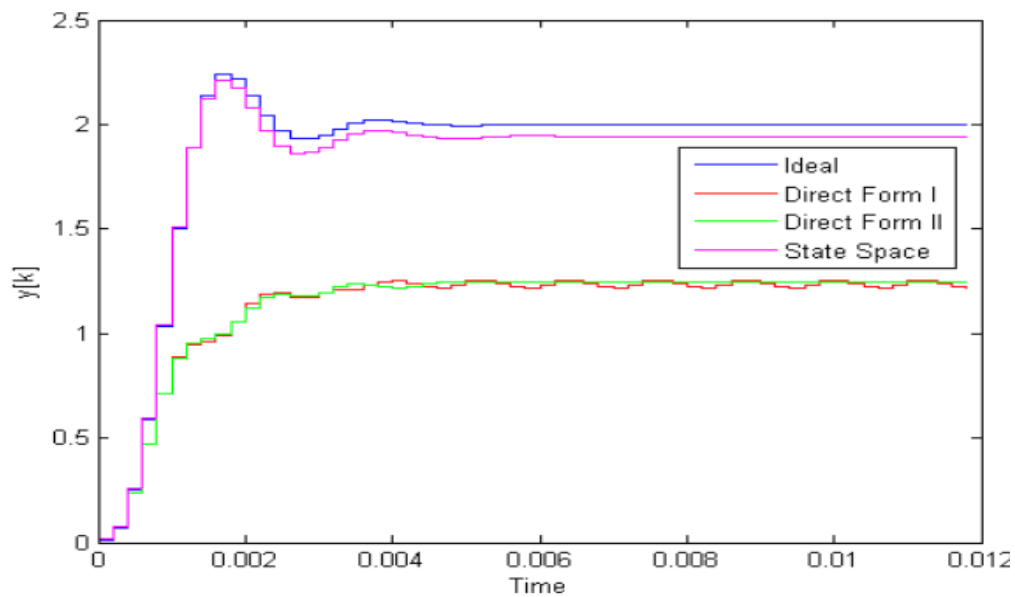


Figure 4. Step Response of 4th order low-pass Butterworth filter with fixed point coefficients⁷.

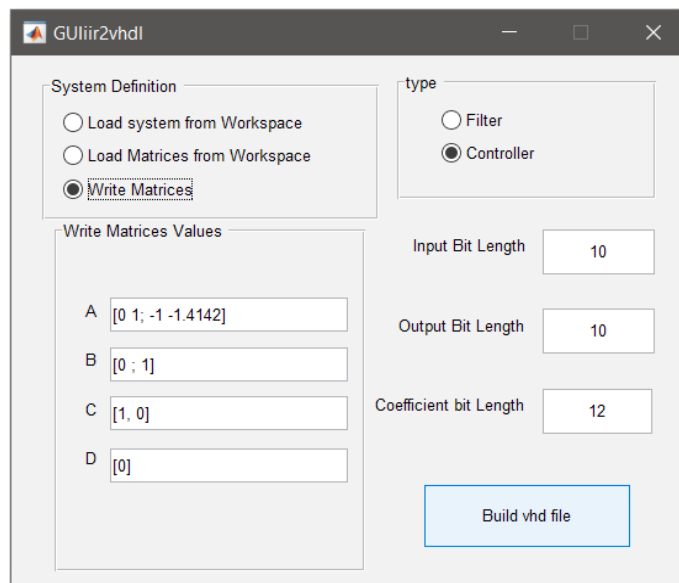


Figure 5. Graphical interface of the tool.

The Graphical User Interface of this tool is shown in Figure 5, it is easy to use, and a VHDL standard code is

generated by entering a minimum of parameters. That code is coherent to the structure shown in Figures 2 y 3.

The interface shown in Figure 5 must be launched from MatLab, it allows the charge from Workspace the transfer function system, the state equation or the state matrices coefficients directly. After this, it must be decided if the code will be generated by a filter or a digital controller. It is also necessary enter the size of the input and output words, according to the converter A/D y D/A used.

4. Results and Discussion

Tests were made with students of digital control and digital signal processing classes from three different universities: Universidad Distrital Francisco Jose de Caldas, Universidad Libre, and Universidad Pedagógica Nacional, located in Bogota, Colombia. For the data collection samples of three years (six terms) were considered. During the three first terms, the tool was not used, but during the last three terms it was. The purpose of this was estab-

lishing a comparison of the academic performance of the students, taking advantage of previous information due to the development of other research works. These were the activities proposed for the students: This project has a deadline that is set during the first classes. It is measured if the student can make the design and implementation of the system at the end of the academic term.

At the beginning of the course rules are given and work groups are set, making clear the academic activities to be developed along the semester, and the final project is defined, having into account that must solve a real problem. The project has to be measured, modeled, designed, simulated, and implemented with a reconfigurable hardware. This taking advantage of the facilities the laboratories of the three universities has available development systems with FPGAs of Xilinx Spartan 3AN for each work group.

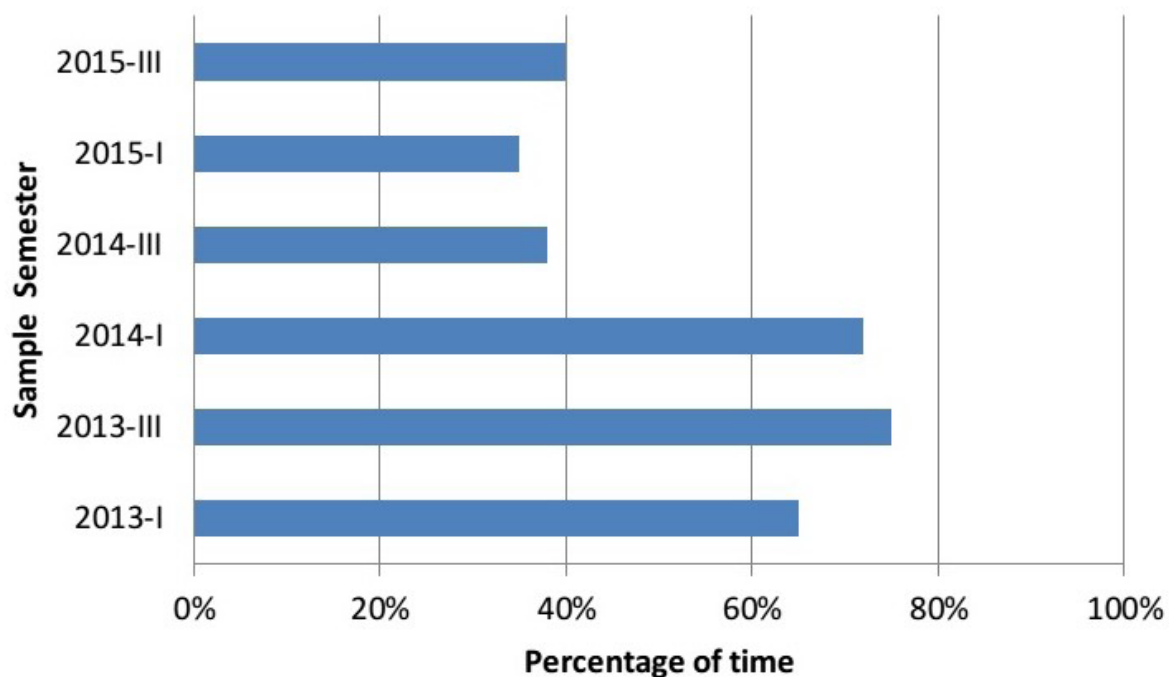


Figure 6. Percentage of time of the implementation task.

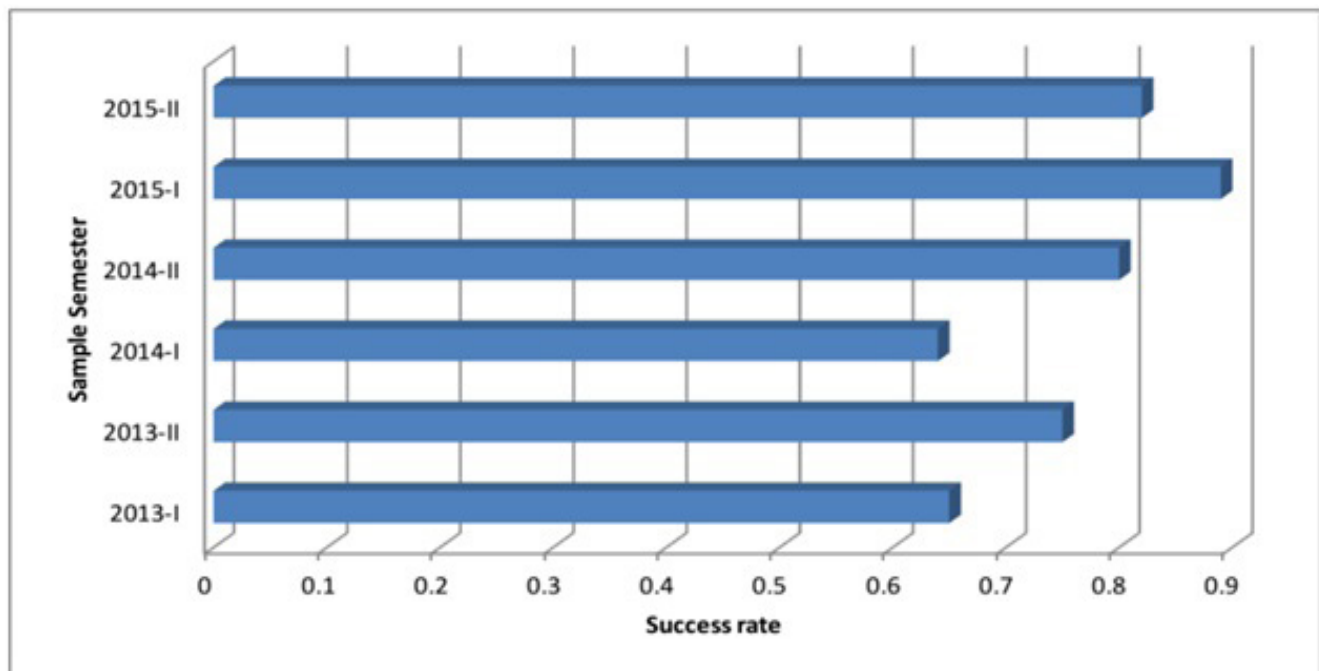


Figure 7. Final project success rate.

The first measurement pattern is the time that students spend in making the project proposed by the teacher. The autonomous work was measured, considering the proportions of the project, and data was divided into the time spent in the technical part of the task, and the time spent until achieving the final implementation. It is important for this research the percentage of time spent by the student in the implementation of the project before and after the usage of this tool. The results are presented in Figure 6. The information shows a decrease from 70% to 37.6% in the time that students spent in programming the device, making the interface with the converters A/D and D/A, in addition to the remodeling of the signals, and the implementation of the power required. This means a reduction of the time allotted for this task in a 53.3%.

The previous indicator is related to the times of the project and its stages, but the time of development was not a guarantee of the project working all the times. The second pattern of interest for both teachers and the differ-

ent curricular projects, it was the number of students who delivered the final project in a successful way that means, how many groups made possible the right functioning of the controller or filter. For doing so, a measurement was made and shown in Figure 7.

Checking this indicator, it is seen an increase in the number of academic success in the results of the student's works. In this case, the percentage of increasing in the number of projects delivered working properly according to the specifications given got bigger in a 23%. Which is meaningful for such a short period. It is important to highlight that those percentages are the result of a measurement in classes of the three universities and different carriers, with an average of the results in each class, without paying more attention to classes with less students, or without verifying variables such as student's schedules, term they are in, number of academic credits, socio-economic conditions, as elements that could affect their academic performance.

According to the data obtained in figures Figure 6 and Figure 7, the influence of the usage of the tool in the classroom represents positive results, especially in the development of the project, when it is necessary to invest an important amount of time in the use the hardware description language and the forms of programming of digital filters.

5. Conclusions

The use of the tool allows that students develop their laboratory practices generating controllers or filters in a high level, which makes possible avoiding potential errors in the implementation. This involves some advantages for the teacher when testing the process since he or she can focus on concepts distinctive of the classes, and not on potential errors in the implementation.

Besides the possibilities in the industrial level, the tool is capable of being used by students of Engineering without knowledge in hardware programming, such as Mechanical Engineering, or Chemical Engineering, among others, which study topics related to control, but not specialized in topics related to implementation.

During the application tests of the tool with the students, a meaningful reduction of time was seen, regarding to the laboratories practice and the class projects. Before the tool, students took longer in the implementation of the controllers in relation to their designs. After using the tool, the times of implementation decreased in relation to the total time of development of the project. This decreasing of time might reconsider the laboratory guides to expand the analysis possibilities, as well as the development of the practices. It would also favor the depth of the content areas.

6. Acknowledgments

This work was supported by the Universidad Distrital Francisco Jose de Caldas, the Universidad Libre and the Universidad Pedagogica Nacional. The views expressed in this paper are not necessarily endorsed by this universities. The authors thank the research groups ALICE,

ARMOS and DETECAL for the evaluation carried out on prototypes of ideas and strategies.

7. References

1. Artigas JI, Barragan L, Isidro U, Navarro D, Lucia O. Fpga-based digital control implementation of a power converter for teaching purposes. 5th IEEE International Conference on e-Learning in Industrial Electronics (ICELIE). 2011; p. 55-60. Crossref.
2. Laakkonen O, Ruama K, Ikonen M, Pyrhonen O. Reconfigurable platform for teaching motor control algorithms. 37th IEEE, PESC '06 Power Electronics Specialists Conference. 2006; p. 1-4. Crossref.
3. Bol EW, Santander ACA, Almeida O, Coehlo L, Coehlo AR. Winfact a tool for teaching digital control. Proceedings of the 38th IEEE Conference in Decision and Control. 1999; 1:353-8. Crossref.
4. Dolgov A, Miao B, Zane R, Maksimovic D. Gui-based laboratory architecture for teaching and research in digital control of smps. IEEE Workshops in Computers in Power Electronics. 2006; p. 236-9. Crossref.
5. Kolek K, Tamau A, Hajduk K, Piatek P, Pauluk M, Marchewka D, Pilat A, Roso M, Gorczyca P. Laboratory real-time systems to facilitate automatic control education and research. Proceedings of the 2010 International Multiconference on Computer Science and Information Technology. 2010; p. 805-12. Crossref.
6. Paiz C, Kettelhoit B, Pörmann M. A design framework for fpga-based dynamically reconfigurable digital controllers. ISCAS IEEE International Symposium in Circuits and Systems. 2007; p. 3708-11. Crossref.
7. Gonzalez ECV, Rivera D, Gomez E. Implementation and simulation of iir digital filters in fpga using matlab system generator. Colombia: IEEE 5th Colombian Workshop in Circuits and Systems (CWCAS). 2014; p. 1-5. Crossref.
8. Rusia P, Bhongade S. Design and implementation of digital pid controller using fpga for precision temperature control. 2014 6th IEEE, Power India International Conference PIICON. 2014; p. 1-4. Crossref.
9. Bhattacharyya A, Sharma P, Murali N, Murty S. Development of fpga based IIR filter implementation of 2-degree of freedom pid controller. Annual IEEE India Conference INDICON. 2011; p. 1-8.

10. Kocur M, Kozak S, Dvorscak B. Design and implementation of fpga - digital based pid controller. 15th International Carpathian Control Conference (ICCC). 2014; p. 233-6. Crossref.
11. Islam S, Sarker R, Saha S, Uddin A. Design of a programmable digital IIR filter based on fpga. International Conference in Informatics, Electronics Vision (ICIEV). 2012; p. 716-21. Crossref.
12. Abbas G, Farooq U, Asad M. Design and fpga implementation of 1-degree-of-freedom discrete pid controller for power switching converter. 7th IEEE Conference on Industrial Electronics and Applications (ICIEA). 2012; p. 1070-4. Crossref.
13. Samanta S, Chakraborty M. Fpga based implementation of high speed tunable notch filter using pipelining and unfolding. Twentieth National Conference on Communications (NCC). 2014; p. 1-6. Crossref.